DEEP TRANSFER LEARNING FOR BRAIN TUMOR CLASSIFICATION USING PRE-TRAINED MODELS, FINE-TUNING AND SVM

A THESIS SUBMITTED TO THE GRADUATE

SCHOOL OF APPLIED SCIENCES

OF
NEAR EAST UNIVERSITY

By

PSHTIWAN JABAR KARIM

In Partial Fulfillment of the Requirements for
the Degree of Master of Science

In

Computer Engineering

NICOSIA, 2021

PSHTIWAN JABAR KARIM

DEEP TRANSFER LEARNING FOR BRAIN TUMOR
CLASSIFICATION USING PRE-TRAINED MODELS,
FINE-TUNING AND SVM

NEU
2021

DEEP TRANSFER LEARNING FOR BRAIN TUMOR CLASSIFICATION USING PRE-TRAINED MODELS, FINE-TUNING AND SVM

A THESIS SUBMITTED TO THE
GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY

By
PSHTIWAN JABAR KARIM

In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Computer Engineering

NICOSIA, 2021

PSHTIWAN JABAR KARIM: DEEP TRANSFER LEARNING FOR BRAIN TUMOR CLASSIFICATION USING PRE-TRAINED MODELS, FINE-TUNING AND SVM.

Approval of Director of Graduate School of Applied Sciences

Prof Dr K. Hüsnü Can Başer

We certify that this thesis is satisfactory for the award of the degree of Master of Science in Computer Engineering

Examining Committee in Charge:

Prof. Dr. Rahib Abiyev	Committee Chairperson, Department of Computer Engineering, NEU
Assoc. Prof. Dr. Kamil Dimililer	Committee, Department of Automotive Engineering, NEU
Assoc. Prof. Dr. Melike Sah Direkoglu	Supervisor, Department of Computer Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Pshtiwan Jabar Karim

Signature:

Date:

To my parents...

ACKNOWLEDGEMENTS

From the bottom of my heart, I want to offer my deepest gratitude and appreciation to many people who sincerely helped me during my study and completing this thesis.

First and foremost, I must express my very profound thanks to the thesis supervisor, Assoc. Prof. Dr. Melike Sah Direkoglu. She tirelessly helped, encouraged, and guided me to have a better understanding and knowledge of Deep Learning to solve the medical image field especially the brain tumor problem.

I also extend my special thanks to the Near East University in general, and the Head of the Computer Engineering Department, Prof. Dr. Rahib Abiyev in specific. His door was always open to help and instruct us, his valuable instructions were unforgettable.

Special and warm thanks to my parents, siblings, and my spouse for their patience, encouragement and help during my study at the university.

Last but not the least, it is necessary to thank my friends who always supported and advised me to obtain this academic achievement.

ABSTRACT

A brain tumor is one of the globally leading reasons for cancer-related deaths in children and adults. The classification of brain tumors is a challenging research issue. With respect to intensity, size, and shape, brain tumors show high variations. Tumors can display similar appearances from different pathological types. To classify and diagnose brain tumors, there are several imaging techniques utilized. Fortunately, due to its superior image quality and the reality of relying on no ionizing radiation, MRI is generally used. With recent developments in deep learning, artificial intelligence (AI) methods can assist radiologists in understanding medical images rapidly. In this thesis, for brain tumor classification, we employ deep transfer learning together with Support Vector Machine (SVM) with a new fine-tuning strategy. First, preprocessing is applied to MRI images. Second, we applied re-sampling as a data augmentation technique. Then, features are extracted from a pre-trained custom CNN model and ResNet-50 model using deep transfer learning. Generally, after the convolution layers, features are flattened and directly given to SVM for classification. On the other hand, in our work, we apply a new fine-tuning of parameters for transfer learning. In particular, after flattening, we apply four consecutive Fully Connected layers with Dropout and ReLU, where the features are also learned during the classification. Then, a Softmax layer is applied, which generates normalized values. Subsequently, the output of the normalized values of the Softmax layer is given to SVM for classification. The efficiency of the proposed deep learning-based classification approach is tested on the Figshare dataset which includes Magnetic Resonance Imaging (MRI) in the three brain tumor sorts; meningioma, glioma, and pituitary. Results show that the proposed deep transfer learning approach is effective and achieves a classification accuracy of 99.61%, which provides better results compared to not using any fine-tuning during transfer learning and equated with other previous work techniques on the Figshare dataset. To the best of our knowledge, for brain tumor classification, this is the first time ResNet-50 based transfer learning together with an SVM classifier has been employed for brain tumor classification on the Figshare dataset. In addition, we also introduce a new fine-tuning strategy during the deep transfer learning stage that improves classification accuracy.

Keywords: Brain Tumor Classification, CNN, ResNet-50, TL, SVM, Figshare Dataset.

ÖZET

Beyin tümörü, çocuklarda ve yetişkinlerde kansere bağlı ölümlerin dünya çapında önde gelen nedenlerinden biridir. Beyin tümörlerinin sınıflandırılması zorlu bir araştırma konusudur. Yoğunluk, boyut ve şekil açısından beyin tümörleri yüksek varyasyonlar gösterir. Tümörler, farklı patolojik tiplerden benzer görünümler sergileyebilir. Beyin tümörlerini sınıflandırmak ve teşhis etmek için kullanılan birkaç görüntüleme tekniği vardır. Neyse ki, üstün görüntü kalitesi ve iyonlaştırıcı radyasyona güvenmeme gerçeği nedeniyle, genellikle MRI kullanılır. Derin öğrenmedeki son gelişmelerle birlikte, yapay zeka (AI) yöntemleri, radyologların tıbbi görüntüleri hızla anlamalarına yardımcı olabilir. Bu tezde, beyin tümörü sınıflandırması için, yeni bir ince ayar stratejisi ile Destek Vektör Makinesi (SVM) ile birlikte derin transfer öğrenmeyi kullanıyoruz. İlk olarak, MRI görüntülerine ön işleme uygulanır. İkinci olarak, veri artırma tekniği olarak yeniden örnekleme uyguladık. Ardından, özellikler önceden eğitilmiş özel bir CNN modelinden ve ResNet-50 modelinden derin aktarım öğrenimi kullanılarak çıkarılır. Genel olarak, evrişim katmanlarından sonra özellikler düzleştirilir ve sınıflandırma için doğrudan SVM'ye verilir. Öte yandan, çalışmamızda, transfer öğrenimi için yeni bir ince ayar parametresi uyguluyoruz. Özellikle, düzleştirmeden sonra, sınıflandırma sırasında özelliklerin de öğrenildiği Dropout ve ReLU ile art arda dört Tam Bağlı katman uygularız. Ardından, normalleştirilmiş değerler üreten bir Softmax katmanı uygulanır. Ardından, Softmax katmanının normalleştirilmiş değerlerinin çıktısı, sınıflandırma için SVM'ye verilir. Önerilen derin öğrenmeye dayalı sınıflandırma yaklaşımının verimliliği, üç beyin tümörü türünde Manyetik Rezonans Görüntülemeyi (MRI) içeren Figshare veri setinde test edilir; meningioma, glioma ve pituitary. Sonuçlar, önerilen derin transfer öğrenme yaklaşımının etkili olduğunu ve% 99.61'lik bir sınıflandırma doğruluğuna ulaştığını, bu da transfer öğrenimi sırasında herhangi bir ince ayar kullanmamaya kıyasla daha iyi sonuçlar sağladığını ve Figshare veri setindeki diğer önceki çalışma teknikleriyle eşit olduğunu göstermektedir. Bildiğimiz kadarıyla, beyin tümörü sınıflandırması için bu, Figshare veri setinde beyin tümörü sınıflandırması için bir SVM sınıflandırıcısı ile birlikte ResNet-50 tabanlı transfer öğrenmesi ilk kez kullanıldı. Ek olarak, derin transfer öğrenme aşamasında sınıflandırma doğruluğunu artıran yeni bir ince ayar stratejisi de sunuyoruz.

Anahtar Kelimeler: Beyin Tümörü Sınıflandırması, CNN, ResNet-50, TL, SVM, Figshare Veri Seti.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI:	Artificial Intelligence
ML:	Machine Learning
DL:	Deep Learning
TL:	Transfer Learning
NN:	Neural Network
CNN:	Convolutional Neural Network
DCNN:	Deep Convolutional Neural Network
PTN:	Pre-Trained Network
Resent:	Residual Network
ILSVRC:	ImageNet Large Scale Visual Recognition Challenge
VGG:	Visual Geometry Group
ANN:	Artificial Neural Network
RNN:	Recurrent Neural Network
GA:	Genetic Algorithm
SVM:	Support Vector Machine
KNN:	K- Nearest Neighbor
SOM:	Self Organization Map
BPNN:	Back-Propagation Neural Network

BT:	Brain Tumor
MRI:	Magnetic Resonance Imaging
CT:	Computer Tomography
FCM:	Fuzzy C-Means
FKM:	Fuzzy K-Means
CAD:	Computer-Aided Diagnosis
BOW:	Bag of Words
GLCM:	Gray Level Co-occurrence Matrix
DWT:	Discrete Wavelet Transform
ELM:	Extreme Learning Machine
GAP:	Global Average Pooling
RBF:	Radial Basis Function
GBUs:	Graphic Processing Units
SAE:	Stacked Auto-Encoder
DBM:	Deep Boltzmann Machine
DBN:	Deep Belief Network
LSTM:	Long Short-Term Memory
NLP:	Natural Language Processing
ReLU:	Rectified Linear Unit

CHAPTER 1

1.1 INTRODUCTION

The brain in the human body is the management center. It is responsible to execute all processes through a huge number of neurons and many connections. The early phase of tumor diagnosis depends on the experience and physician’s knowledge, making the patients have a chance to resume his life and survival (Gumaei et al., 2019); as in Figure 1-1. The most devastating disease is brain tumors, heading to a very short life hope in their highest level. There are various kinds of brain tumors that can either be benign or malignant.

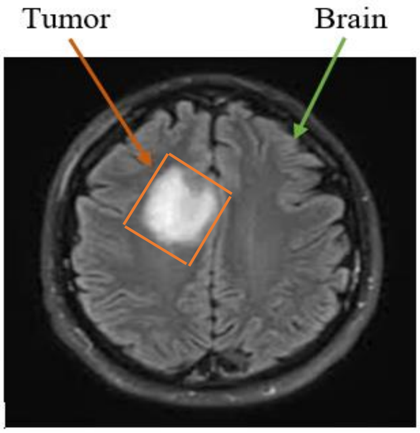


Figure 1-1: MRI Brain Tumor.

Benign tumor: is a non-progressive (non-cancerous) form, and it originates in the brain and gradually developing. This kind of tumor will not be spread anywhere in the body, and it is considered to be less aggressive. Tissue or portion of the brain which can be taken off on time, and pressed via the irregular growth of cells.

Malignant tumor: It is a progressive (cancerous) type. It breaks away rapidly through unknown borders, invades other normal tissues, and spreads to the other areas of the human body. If the brain is the center of this kind of tumor, then it is defined as a primary malignant tumor. When it emerges in the other parts of body, it extends to the brain. It is also recognized as a secondary malignant tumor. (Sultan et al., 2019); as illustrated in Figure 1-2.

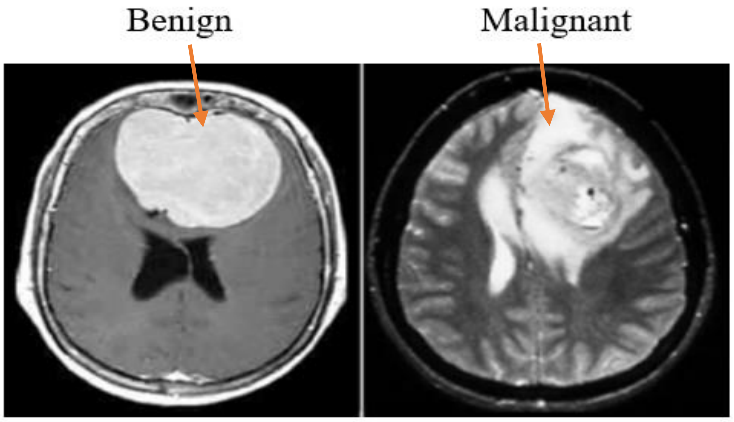


Figure 1-2**:** Benign and Malignant Brain Tumors.

Brain tumors can be categorized into two classes, including primary and secondary. The primary accounts for approximately 70% of all tumors of the brain, while the remaining 30% are secondary tumors. This category is defined by the origin of the tumor; just as primary tumors are considered tumors that first originate in the brain. On the other hand, within Malignant its first tumor called primary appears in some other part of the body and then, it is changed to its secondary tumor that is moved to the brain, and both of them are malignant.

In 2015, in the USA, approximately 23,000 patients were diagnosed with brain tumors. A brain tumor is estimated as one of the main causes of cancer-related sickness, mortality, and morbidity globally. According to 2017 cancer statistics, the brain tumors found in both children and adults (Rehman et al., 2020).

The most significant kinds of brain tumors are meningioma, glioma, and pituitary:

Meningioma: It is the most common type of benign tumor that instigating the soft membranes that cover the spinal cord and brain.

Glioma tumors: are number of tumors that develop inside the brain compound. The high-grade glioma is one of the most dangerous brain tumors with at least survival of approximately two years.

Pituitary tumors: brain cells abnormally become large. In this sort of tumor, the gland of the brain grows as well. These tumors are similar in shape, inherent and nature. The spread in any place in the brain (Abir et al., 2018).

The most significant variation among these three kinds of tumors is that meningioma are usually benign, while gliomas are most often malignant. Pituitary tumors, even if they would be benign, they can cause so many other medical issues, unlike meningioma tumors which are slow-growing. Since of the details described above, the precise distinction among these three kinds of tumors represents a very significant phase in the clinical diagnosis process and later impressive evaluation of patients (Badža & Barjaktarović, 2020). Figure 1-3 Show the three different type of brain tumors.

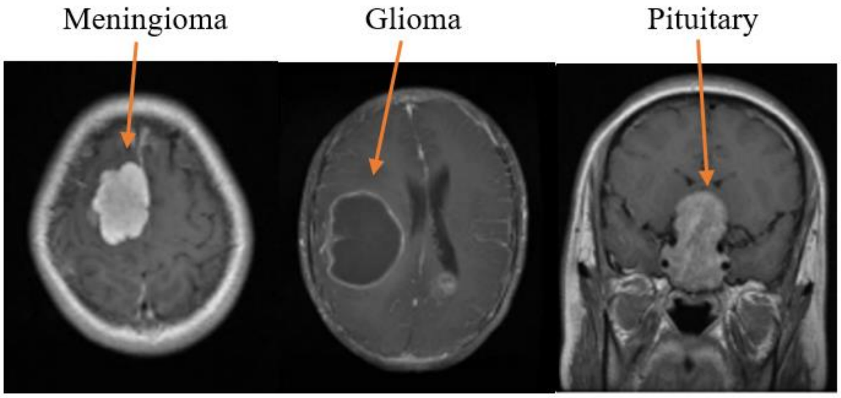


Figure 1-3**:** Brain Tumor Types.

Brain tumor image testing is conducted by using x-rays and powerful magnets, or radioactive substances to generate brain images. Brain tumors are usually diagnosed with using several kinds of scans including of Magnetic Resonance Imaging (MRI), Computer Tomography (CT), Emission Tomography Myelogram (ETM), Positron, and Angiogram are among the kinds of scans that are used mostly to diagnose brain diseases. These images are so effective that they are able to provide primary information about the tumor's location and existence of brain tumor classifications even among subtypes as research challenge problems. To identify, segment, and classify brain tumors, various imaging methods can be used. However, one of the most prevalent techniques that is non-invasive MRI. The success of MRI comes from the ability to use no ionizing radiation during the scan, as well as its better resolution of thin tissue. In addition, they have capability to obtain various images apply different image parameters, or using contrast-enhanced factors (Litjens et al., 2017).

For the detection, classification, and segmentation of brain tumors, several techniques have been proposed. In the area of medical imaging, Machine Learning (ML) has appeared widely as a subclass of Artificial Intelligence. ML is the analysis of the statistical, algorithm, and mathematical equations that can be used to perform a particular task instead of focusing on patterns without using straightforward instructions. It is possible to break ML into four main classes, supervised, unsupervised, semi-supervised, and reinforcement learning. An algorithm is used to supervise techniques, detect a mapping function of input variables and their corresponding output labels to predict novel subject labels. The major aim of this is to understand inherent patterns within the training data using models into K-Nearest Neighbors (KNN), Artificial Neural Network (ANN), and Support Vector Machine (SVM). Unsupervised learning, on the other hand, is based only on input variables in clustering such as Fuzzy C-Means (FCM) or Fuzzy K-Means (FKM) and Self Organization Map (SOM) (Sultan et al., 2019).

Deep Learning is most commonly used for the analysis of brain images in many applications includes normal and abnormal brain tumor detection, classification, and segmentation (non-enhancing tumor zone, enhancing and edema), stroke lesion segmentation, Parkinson, Alzheimer, and brain tumor diagnosis… etc. (Chahal et al., 2020). DL is a kind of artificial neural network-based ML in which multiple processing layers are used to gradually extract higher-level of data elements that help to overcome several challenges that occur in traditional machine learning methods (Tandel et al., 2019).

On a huge dataset, DL is a predicted model. DL is often functioned in medical imaging to recognize damaged sections of any object, it is lastly affected parts of the lungs, and it is also helpful to categorize the images and prediction techniques of the object (Noreen et al., 2020). In different ways, Computer-Aided Diagnosis (CAD) programs have assisted neurologists. Additionally, neurological CAD applications support tumor classification, grading, segmentation, and detection (Deepak & Ameer, 2019). A Convolutional Neural Network approach including a classifier and a feature extractor acts as a combined unit. Actually, there is a significant interest in using CNN to develop CAD systems. The CAD systems which has used CNN it has been extremely successful and obtained remarkable outcomes (Deepak & Ameer, 2020). DCNN has been developed to resolve the above-mentioned deficiencies, but when the data size is small, they begin to overfitting. To amend this error of overfitting, transfer learning concepts with pre-trained DCNN techniques and data augmentation are developed (Talo et al., 2019). Transfer Learning is another concept of deep learning models to deal with performance issues. These tasks gained from the prior models are applied this information to another domain. Thus, if we have a small dataset, this technique is really very important. When the data number is relatively small, after many epochs, the model begins to over-fit. If the previous dataset is huge general and enough, the learned features can be applied to categorize various classes that do not exist in our original dataset. Another benefit of TL is that there is no need for high computational power (Kaur & Gandhi, 2020). In this thesis, we use deep learning methods for the classification of brain tumors. A figshare dataset including MR images for the various kinds of brain tumors (Figshare Dataset): Meningioma, Glioma, and Pituitary have been applied. We used CNN and Resnet5 with TL scenario to extract features for the standard MRI images of the brain structure. The suggested SVM image classifier was evaluated and compared with several other known techniques of classification.

1.2 Aim of the Study

The purpose of the thesis is to develop automated techniques to help doctors in diagnosing process in order to avoid misdiagnosis and reduce waiting time for patients. In particular, this work achieves this automation through the classification of three different types of brain tumor MRI images. To identify health problems that take time away from more complicated diagnoses, images enable a doctor to review multiple image slices. Our aim is to confidentially categorize various kinds of brain tumors to reduce the burden of doctors, leaving them with the most complex diagnoses. We used CNN and Resnet5 with TL scenario to extract features for the standard MRI images of the brain structure. The suggested SVM image classifier was evaluated and compared with several other known techniques of classification. It was observed that SVM outperforms in the context of accuracy when compared to the existing algorithms.

1.3 Significance of the study

Due to Machine Learning and Deep Learning, significant developments have been made in medical science over the past few years, such as the medical image processing technique that allows doctors to diagnose the disease earlier and more rapidly, the disease were tedious and time-consuming before the invention of these techniques. Computer-aided technology is therefore much needed to overcome such constraints because the medical field needs accurate and efficient methods to diagnose life-threatening diseases such as cancer, which is the world's main cause of mortality for patients. Thus, we provide a model for the classification of various types of brain tumors using the convolutional neural network and ResNet-50 models with support vector machine, transfer learning in our study with the aid of Brain MRI Images.

1.4 Limitations of the Study

There are some limitations that need to be considered when dealing with the classification of brain tumor types:

- Firstly, the shortcoming of these schemes is their binary classifier of the tumor, which leaves the radiologist with several ambiguities. The explanation is that categorization into benign and malignant is not necessary for the radiologist to determine the patient's treatment and prevention. The categorization necessarily to be multi-class, which classifies brain tumors to their respective classes, to get a better and clear understanding of radiologists. In addition, the absence of data is also a key obstacle for researchers to obtain accurate outcomes.
- Secondly, the brain tumor categorization according to its sub-type is another significantly related issue. Brain tumors of the identical category may have differences based on various patient-specific agents in structure, size, and shape. Tumors from different categories, on the other hand, could display similarities in presence. This act makes the issue more complex. In contrast, limited studies have been published to categorize brain tumors into various pathological sorts.

1.5 Problem Statement

A brain tumor is a lethal disease in the medical area, due to the heterogeneous existence of the tumor cells, its classification is difficult to work for radiologists.

The classification of brain tumors is divided into two categories:

- MRI classification into a normal and abnormal tumor.
- Classification into various tumor kinds within abnormal brain tumors.

Automatic brain tumor classification into different pathological kinds multi-classification into (meningioma, glioma, and pituitary) It is a hard issue in a contrast to the binary categorization of tumors into normal and abnormal. In this thesis, we use DL methods to solving the brain tumor multi-classes problem.

1.6 Methodology

The purpose of this thesis is to improve the accuracy of the classification of brain MRI images by applying DL methods, ML algorithm, and the approach of Transfer Learning (TL). TL is the task of using the knowledge given by a pre-trained framework to learn new models provided by new data. It's typically simpler and much easier to calibrate a pre-trained system with TL rather than starting from basic. The use of pre-trained DL systems gives us the ability to learn new works quickly. Here, we review the two distinctive DL models such as CNN and ResNet-50 using brain tumor classification MRI images and applying TL techniques to the given dataset. Pre-trained CNN and Resnet-50 models are used to execute TL to eliminate visually important and extraction of features. Finally, using the Support Vector Machine, the classification of these features is completed. It starts with the image dataset of figshare, which was collected and arranged into three distinct kinds of BT like Meningioma, Glioma, and Pituitary (Figshare Dataset). The proposed strategy involves the following stages: first, preprocessing every MRI image, second, applied re-sampling as a data augmentation technique, third, extraction of features based on DL and TL, finally the classification of various types of brain tumor by using SVM.

1.7 The Study region and data

Figshare is the only available dataset for the three particular tumor types discussed in this study. The figshare dataset is freely accessible and is widely used to test algorithms for retrieval and classification. This is a series of 3064 brain MRI images were obtained from 233 patients who were diagnosed with one of the brain tumors kind such as meningioma, glioma, and pituitary. The images relate to the modality of T1-CE MRI and contain views of coronal, sagittal, and axial. Includes 708 brain MRI images of the meningioma images (corresponding to 82 patients), 1426 glioma (89 patients), and the remaining 930 images relate to cases of the pituitary (62 patients). The images are accessible as mat files and each image's size is 512x512 (*Figshare brain tumor dataset*, 2018).

1.8 Overview of the Study

The thesis includes the following chapters to achieve the brain tumor classification system design.

Chapter 1: This is an introduction to the topic of the thesis. The summary of the study is outlined, which describes the purposes, significances, limitations, problem statement, methodology, study region, and the data of this thesis.

Chapter 2: Through reviewing the use of the techniques of machine learning a long with methods of deep learning, and transfer learning concerning the classification of medical images. Especially, brain tumor types classification. The discussions provide some relevant research works presented to solve the brain tumor issues by using machine learning algorithms and deep learning models.

Chapter 3: Briefly indicates a summary about machine learning and Support Vector Machine as an algorithm. This chapter also comprehensively and in detail talks about deep learning methods such as Convolutional Neural Network, ResNet-50, and Transfer Learning.

Chapter 4: Explains the basics of the methodologies that contains designing a system to classify various types of brain tumors. It also explains the proposed method, preprocessing, feature extraction, and classification.

Chapter 5: Represents the experimental result analysis of the two models that each one has its own outcomes separately. Meanwhile, the obtained results for each model are compared together and then with the results that have been achieved by the other researches in the field of brain tumor classification.

CHAPTER 2

2.1 LITERATURE REVIEW

The classification of brain tumors separates the image to assess the tumor's region and extracts quantitative features such as size, shape, texture, and intensity. But this approach requires advanced information of the sort of feature extraction. For the classification of brain tumor diagnosis and segmentation, various methods and approaches have been suggested; including Machine Learning algorithms and Deep Learning techniques. In several fields like medical diagnostic, ML has widely functioned. However, few researches, specifically using MRI imaging that have targeted brain tumor diagnosis. ML models train and test traditional ML methods for MRI images mostly. Newly deep learning for brain tumor diagnosis has been applied by some methods. With the progress of computer vision and deep learning, researchers start to use CNN for the diagnosis of brain cancer classification and segmentation. CNN requires no previous information of feature types. And also can be trained end-to-end without the tumor images been segmented (Noreen et al., 2020) (Liu et al., 2019) (Cheng et al., 2019).

Several models and solutions for the classification of brain tumor diagnosis applying MRI images that they have been presented by some authors in the last four years. These techniques consist of image processing, traditional ML algorithms, and DL models.

The first and important work of the brain tumors classification including meningioma, glioma, and pituitary, based on the figshare MRI images (Cheng et al., 2015). Three methods for extraction of features were used, such as intensity histogram, Bag of Words (BoW), and Gray Level Co-occurrence Matrix (GLCM). The aforementioned researchers compared the classifiers by intensify tests they reached good results in diagnosing the brain tumor regions. The best result was obtained by combination of BoW features and an SVM classifier. Five-fold cross-validation was followed by experiment assessment and overall accuracy was obtained is 91.28%.

(Ismael & Abdel-Qader, 2018) Presented a model for the classification of brain tumors in MR images that incorporates neural network algorithms and statistical features. They employed feature extraction of brain MRI by using a combination of the Two-Dimensional Gabor filter technique and Two-Dimensional Discrete Wavelet Transform (DWT) algorithm. The authors Applied multilayer perceptron neural network trained (back-propagation neural network) for classification. A large dataset including 3,064 images of T1-weighted MRI of the three sorts of brain tumors such as meningioma, glioma, and pituitary tumors were used. This work obtained an accuracy result which is 91.9%.

Another study by (Afshar et al., 2018) described a Capsule network (CapsNet) model for the classification of brain tumors based on four objectives that consists of incorporate and adopting Capsule Network, over-fitting analysis, development of visualization model for production, and also the capability of cabinets. The research showed an announced accuracy of 86.56% in the convolution layer by using Capsule Network.

(Pashaei et al., 2018) in their study suggested that a CNN technique was created to feature extraction from brain tumor MRI images. There were five learnable layers in the model and the filters had the size of 3x3 for all the layers. The CNN method claimed to obtain a 93.68% accuracy of classification. applying CNN features with a classifier process from the extreme learning machine (ELM) class, the performance was enhanced. Within this research, recall measures were very high for the class of pituitary tumors, while they had a very low measures in meningioma. This proposes restrictions in capability of the classifier to discriminate.

One more research by (Abiwinanda et al,. 2019) used a DL model based on the Convolutional Neural Network that applied the three most famous classes of brain tumors as meningioma, glioma, and pituitary for the classification. A construction consisting of convolution, flattening layer, max pooling, and the fully connected layer from a single hidden layer came after them. And they were involved in the method. The validation and training accuracies were obtained from the analysis are 84.19% and 98.51%.

(Anaraki et al., 2019) Presented a design based on Genetic Algorithm and Convolutional Neural Network to classify various kinds of glioma grades employing MRI images dataset. GA was used in the presented model to automatically pick the CNN architecture. In one research work, they briefly discuss the 90.9% accuracy that was achieved for categorizing three grades of gliomas. Meanwhile, the meningioma, glioma, and pituitary tumor classes were classified with an accuracy of 94.2 % in another research study.

(Deepak & Ameer., 2019) They engaged pre-trained GoogleNet to the feature's extraction from figshare MRI images. Also applied the idea of the transfer learning of Deep CNN method with its softmax, as a stand-alone model, Support Vector Machine (SVM) Deep CNN features, and K-Nearest Neighbor (KNN) classifiers for brain image classification. In this research they presented a system obtained an accuracy level of 98%.

Another research of (Deepak & Ameer., 2020) demonstrates that they employed convolutional neural network features. And a support vector machine is suggested for the classification of brain images. For the three different kinds of brain tumors (meningioma, glioma, pituitary) the fully automated method is tested by functioning the figshare available dataset including magnetic resonance imaging. CNN is structured to extract features from MR images of the brain. A multi-class SVM is operated with CNN features to improve a better performance. The classification's result accuracy 95.82 % was achieved by the proposed system.

(Liu et al., 2019) Presented a novel system named Global Average Pooling ResNet-34 for brain tumor classification. Their system has the following features: the implementation of the established CNN model for the classification task in the field of deep learning called ResNet34, to decrease the number of parameters and to prevent overfitting. Instead of the flattened layer for the classification, they used global average pooling layer. They concatenate the feature vectors of various layers in order to be capable to fuse the low-level and high-level features of the network to increase the accuracy of categorization. Furthermore, they introduced a loss function, which is the sum of the cross entropy loss and the interval loss. The sum total loss increases the punishment for misclassification. In this work, the system obtained the classification accuracy of 95.00%.

(Swati et al., 2019) Have offered the use of deep TL for the automated classification of brain tumors. In this work, researchers experimented various pre-trained networks, such as (Alex-Net, VGG-16, and VGG-19). The architecture using VGG networks obtained better accuracy compared to that Alex-Net. They used pre-trained VGG19 for the brain tumors classification of various layers in the network. The block-wise fine-tuning process resulted in an inaccuracy of up to 94.82%.

For brain tumor identification (Toğaçar et al., 2020) presented a new CNN model called BrainMRNet. In each image in this model, they used a pre-pressing technique with extracted features and data augmentation. The BrainMRNet model is more effective in this work than the deep convolutional neural network models (AlexNet, GoogleNet, VGG-16) pre-trained. With the BrainMRNet model in this work, the classification performance achieved was 96.05%.

(Talo et al., 2019) Suggested an approach to automatically classify normal and abnormal MRI brain images by using deep transfer learning. As a deep learning method, the CNN-based pre-trained (ResNet34) network is used. In the dense layers of the model, the authors proposed modifications, fine-tuning, and data augmentation with training the model. The suggested framework obtained classification accuracy of 100% on 613 MRI images.

(Ghosal et al., 2019) Presented an automation approach from MRI data for the brain tumor classification of different types into meningioma, glioma, and pituitary where the image slice samples are transferred to a Convolutional Neural Network (CNN)-based Squeeze and Excitation ResNet model. They used data augmentation to further enhance in the performance. The accuracy of 93.83% was achieved in this study.

(Rehman et al., 2020) Introduced a system for brain tumor classification into Meningioma, Glioma, and Pituitary, then they used three different types of Pre-Trained Networks such as (AlexNet, GoogLeNet, and VGGNet16). The algorithm that referred to the above sliced brain MRI to find regions of interest. In this study, the researchers have examined data augmentation approaches to achieve better accuracy outcome. They used the Transfer Learning (TL) scenarios that are fine-tuning and freezing. Beside using the TL concepts, they used these pre-trained networks to dig out features. Finally, the SVM and log-based softmax layer have been achieved for the classification of features. Compared to AlexNet and GoogLeNet, they have used a fine-tuned VGG16 network, their experimental approach on the categorization of BT achieved the best accuracy of 98.69%. In the freeze model of TL, on the other hand, the highest accuracy of 95.77% used AlexNet's freeze Conv5 layer as opposed to its other layers, as well as all VGG16 and GoogLeNet architectural layers.

CHAPTER 3

MACHINE LEARNING AND DEEP LEARNING

3.1 Machine Learning

Machine learning is a subclass of artificial intelligence that its goals are to allow machines to use intelligent software to perform their jobs skillfully. The approaches of statistical learning form the backbone of smart software that is used to build machine intelligence. Since machine learning algorithms need data to learn, the discipline must have a relation with the database training. Similarly, concepts like Information Discovery from Data, pattern recognition, and data mining are familiar. One wonders how to see the huge picture in which it illustrates such a connection (Mohammed et al., 2016).

Approximately one out of the two patients with cancer require radiotherapy (ionizing radiation) as a section of the therapy, it's a primary therapy procedure at the progression grades of the sickness. X-ray therapy requires an enormous collection of procedures that not only cover the duration from consultation to therapy but as well as reach further, to make sure those who receive treatment obtained the required radiation dose and they respond properly. The sophistication of these procedures can differ and may require many phases of advanced individual-machine interactivity and making decision, that will normally enable the utilization of ML methods to automate optimization procedures, consisting of not narrowing quality assurance of radiation, treatment planning, contouring, respiratory motion management, image-guided radiotherapy, outcomes prediction, and treatment response modeling.

An effective algorithm has two special important advantages. First, it can replace human efforts that are laborious and repetitive. Secondly, and more importantly, it is possible to get more complex and nuanced patterns in the input data than can be achieved by the ordinary human observer. Both of these benefits are essential to radiation therapy. For instance, during care preparation, the regular contouring of organs and tumors at danger is a time-consuming pattern recognition process that is dependent on the intimacy of the knowledge and observer with the appearance of anatomy in image diagnosis. However, this familiarity has its limits, and thus, in the resulting contours, there is ambiguity and inter-observer variability. It is possible that an algorithm for contouring can at the same time integrate data from several shapes or sources in one image or raise subtleties of texture or thus lessen the uncertainty in the contour, and combine the experience of many observers (Naqa & Murphy, 2015).

3.1.1 Machine Learning Algorithms

ML can be separately depending on the essence of the data labeling into supervised, unsupervised, semi-supervised, and reinforcement learning; as shown in Figure 3-1.

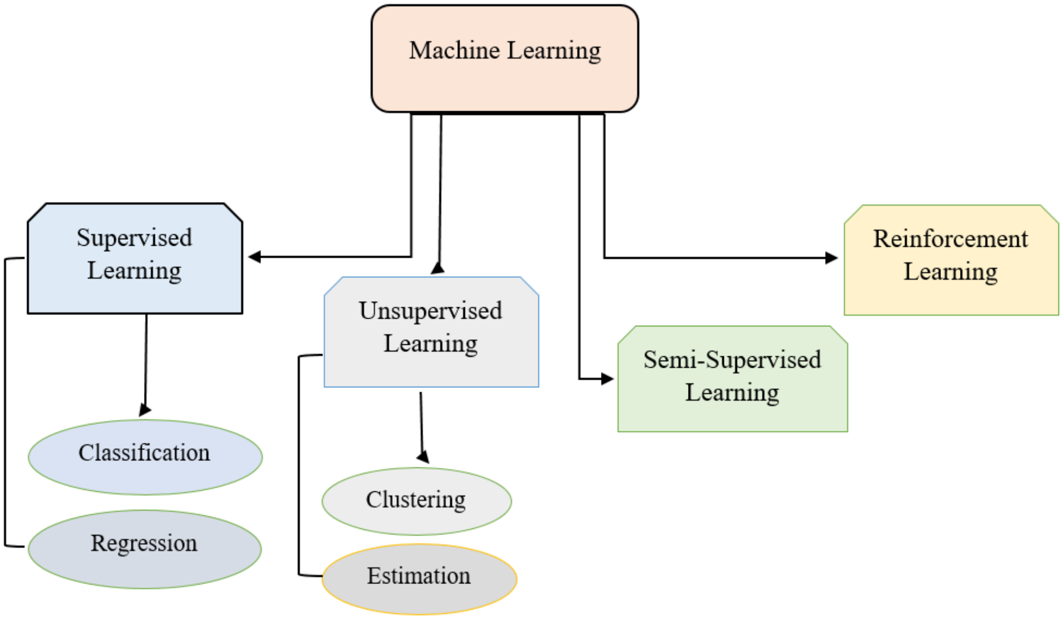


Figure 3-1: Machine Learning Algorithms Diagram.

3.1.1.1 Supervised learning: It is applied to evaluate an anonymous into the input and output mapping known into the input and output examples, where the output is labeled, such as regression and classification. We applied the Support Vector Machine as an algorithm that uses for classification in our thesis, to classify the three different types of brain tumor.

3.1.1.2 Unsupervised learning: It is justly regular in classification issues. Here, only input examples that are granted to the learning model. For instance, probability density function estimation, and clustering.

3.1.1.3 Semi-supervised learning: This sort of learning contains a connection of the two supervised and unsupervised. There a section of the data is labeled and other parts are unlabeled, it is used to produce a suitable model for data classification. In this design, the labeled portion can be applied to assist the learning of the unlabeled portion. This type of model imparts itself to most activities in nature and more accurately imitates how humans improve their expertise. The aim of this learning type is to understand a system which would forecast classes of upcoming test data improvable than the system produced by applying the labeled data only (Mohammed et al., 2016); as shown in Figure 3-2.

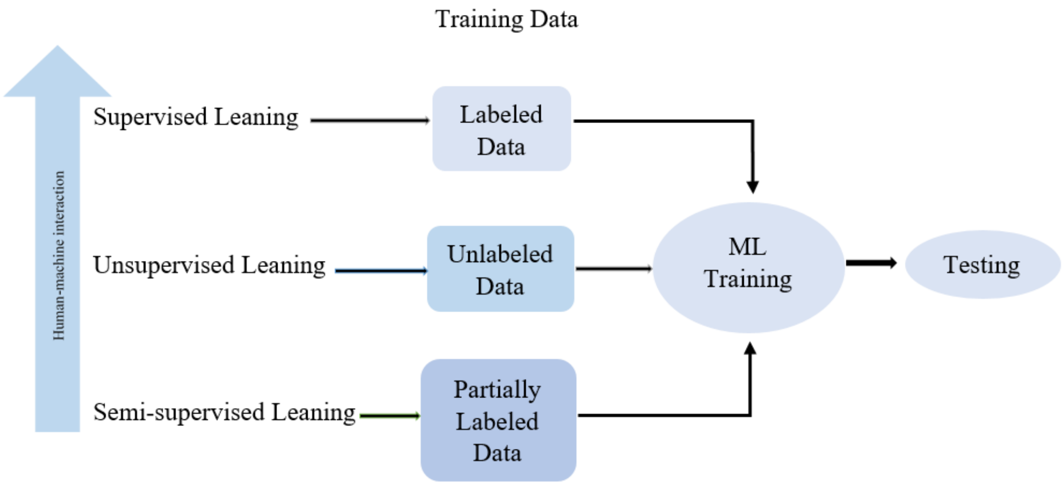


Figure 3-2**:** Training data nature for machine learning algorithm types.

3.1.1.4 The reinforcement learning: is the process that has goals at applying observations collected in the contact along with environment to do actions which will decrease risks or increase rewards. For generating smart systems that is named agents. The following necessary process goes through reinforcement learning: In the first process, the Input state is observed by the agent. In the second process, the function of decision-making is applied for creating the agent that performs an action. In the third process, after the action is executed, the agent gets reinforcement from the environment or reward. Finally, the state-action couple knowledge about the reward is saved. (Mohammed et al., 2016). It has shown in Figure 3-3.

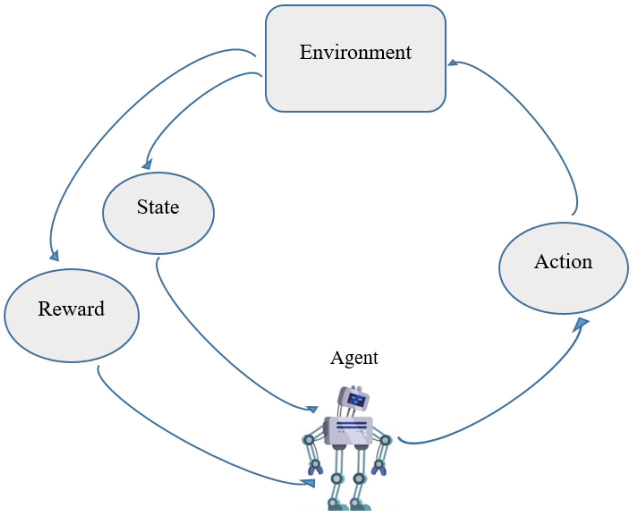


Figure 3-3: Reinforcement Learning Process.

3.2 Deep Learning Method

Deep Learning (DL) methods are a subfield of Machine Learning, but the major benefit of DL over ML is that from the raw images it can automatically extract relevant features. The DL models have been taken to the forefront of Artificial Intelligence (AI). For years, connecting models have existed such as neural networks, but new architectures and efficient machines for computing Graphics Processing Units (GPUs). In the classical method, before the learning of the classification algorithm begins, features are extracted and must be specified by the creator of the algorithm. In the DL method, the classification algorithm itself learns and defines features based on patterns and structures in the training data. The algorithm researches for significant data features that can assist in the decision-making process. One of the goals of DL is to replace automatic feature extraction with hand-crafted features, unsupervised, and semi-supervised learning. DL is not a single algorithm, it contains a group of topologies and that can algorithms be applied on a wider scale to the problem. Several DL algorithms have been used in medical image processing, such as Deep Stacking Network (DSN), Deep Boltzmann Machine (DBM), Deep Belief Network (DBN), Stacked Auto-encoder (SAE), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM)/gated recurrent unit are some of the most famous DL models. For image, video, and natural language processing (NLP), CNN is the most popular and widely used model. Also, since of its self-learning features, CNN is the foremost and first selection of medical image fields (Tandel et al., 2020).

In the medical field, deep learning methods are very significant key and proven useful tools in various important diseases, including the image analysis of breast cancer, diagnosis of lung disease, and brain tumor detection and classification. DL is a sub-type of ML, has become the center of appeal since it is able to present an efficient expecting model by applying comprehensive data into text and images. From a part of it, anticipating the model on huge datasets, DL is able to provide the concluded outcomes. Deep learning is often applied in medical imaging to recognize harmed regions of any part of the body such as the damaged part of the lungs, and it is also helpful to model for predicting and identifying images of objects. To recognize the various patterns in cell images, the assessment with DL has an important repayment. The data analysis and accuracy of prediction systems through DL methods depend primarily on the data sample and its training, as better results require more precise data (Noreen et al., 2020).

Transfer Learning is another way for Deep Learning models to deal with performance issues. Representations gained from the prior model that are used and this information is applied to another domain. Thus, if we have a small dataset, this technique is really very important. When the data number is relatively small, after many epochs, the model begins to over-fit. If the former dataset is huge general and enough, the learned attributes could be applied to categorize various classes which do not have in our first source dataset. The other profit of TL is what that does not require for a high powerful computation (Kaur & Gandhi, 2020).

The motivation to apply pre-trained DL is time-saving since it does not need a huge data set to achieve outcomes. These systems as well as extracted random features from the classification of images. The top layers extracted lower level features including edges, texture, and color. The bottom layers of the systems extracted high-level features like contours and objects. Writings, typically extracted of features by applying pre-trained systems are from bottom layers as the features on top layers of pre-trained systems are approximately similar in medical images and natural. The bottom layers' features would be various from medical images to natural images. The primary goal is to features are extracted from unlike layers of pre-trained systems trained on our presented dataset that are composed to extract multi-scale information from input images to better raise the feature ability of the model of the classifier. AI, ML, and DL are illustrated in Figure 3-4.

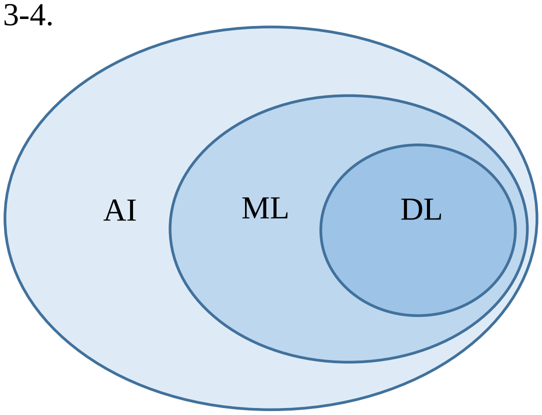


Figure 3-4: AI, ML, and DL Introduction.

3.2.1 Neural Network

Neural networks (NNs), often named Artificial neural networks (ANNs) as well, they are statistical learning techniques motivated by the human brain. NNs include a collection of simple artificial neurons that are combined together to build a network. These relations are described by adaptive weights, which are tuned in the course of the process of learning It is shown as in Figure 3-5. NN was commonly applied for classification works, but due to the computational requirements of NN, they were eventually substituted by simpler methods such as SVM. After the implementation of the DL process and its performance in many speech recognition problems and universal image, their amicability began to minimize again. Since one of the methods of the DL, especially Convolutional Neural Networks (CNNs), is applied in this work, and more detailed explanation will be given below.

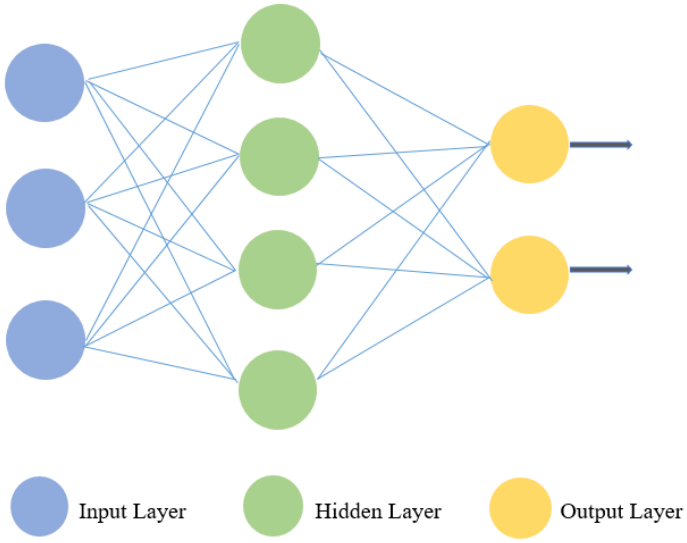


Figure 3-5: Neural Network Architecture.

3.2.2 Convolutional neural network

CNN is a grouping of ANNs which being overcame in different computer vision duties. It gets pay attention among a diverse field, consist of using radiant energy in medical diagnosis and therapy (radiology). CNN is constructed to automatic manner and adapt to acquire information special hierarchy of elements with back-propagation by applying many layers like convolution layers, pooling layers, and fully connected layer. Convolution and pooling layers, conduct extraction of features, while, a fully connected or Dense Layers map the feature extraction, like classification, in the final output. CNN constitute a sheaf of mathematical functions, like convolution operation, it is a specific kind of linear operation and plays a key role. In digital images, pixel values are saved in a 2D grid, for instance, a small grid of parameters and a number of arrays named a kernel, a feature extractor optimization, are added to every image location making CNN highly successful for processing images, as a feature can occur at any place in the image. Extracted features can become more complicated hierarchically and progressively as one layer feds its output into the next layer. The operation of optimizing parameters including kernels is named training, that is carried out through an optimization algorithm named gradient descent and back-propagation, between others, to reduce the dissimilarity between ground truth labels and outputs (Rikiya Yamashita et al,. 2018).

Instead of vanilla neural networks and traditional machine learning, the key profits of CNN are limitless accuracy and feature learning, which can be accomplished by growing training samples and thus contributes to a more stable and precise design (Litjens et al., 2017). The convolutional filters serve as function extractors in the CNN architecture, and be more complex features and extract more as we go deep (structural and spatial information). The extraction of features occurs through combining tiny filters and input patterns, followed by the collection of the most distinguishing features, and then the categorization network begins to be trained (Lecun et al., 2015).

3.2.2.1 Convolution Layer

Convolution is a particular sort of linear function applied to extract of features, there a tiny array of quantities (numbers), named a kernel, is used through the input, that is an array of quantities, named a tensor. An element-wise product among all the features of the kernel and the input tensor is computed at every position of that tensor and added to achieve the output value in the equivalent place of the output tensor, named a feature map. This program is reiterated by using several kernels to constitute an undetermined number of feature maps, that show various elements of the input tensors; variety of kernels can be put in consideration as various feature extractors. Two key hyper-parameters which describe the convolution functioning are the number and size of kernels. The previous is commonly (3 × 3), and some of the times (5 × 5) or (7 × 7). The latter is undetermined and decides the deepness of maps of output features. It is illustrated as in Figure 3-6.

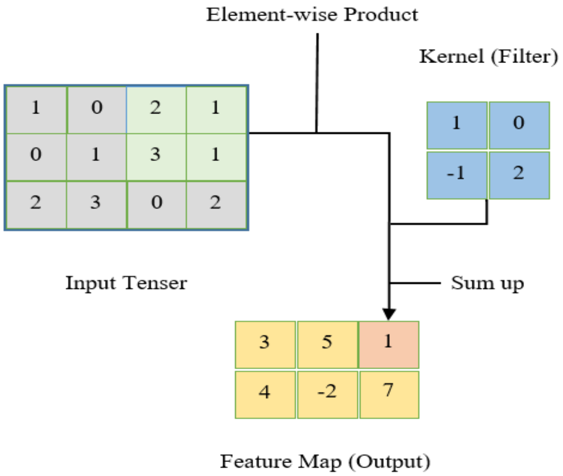


Figure 3-6: Convolution Operation Process.

The convolution operation explained before does not permit the mid of every kernel to overlay the furthermost feature of the input tensor and decrease the width and height of the output feature map to be compatible to the input tensor. To point out this problem, a process is used called padding, or especially zero padding, in which columns and rows of zeros are increased on every side of the input tensor, to suit the center of a kernel on the furthermost characteristic and with the convolution operation to retain the same in-plane dimension. Novel CNN structures typically utilize zero paddings to keep in-plane dimensions so that using more layers. Except zero paddings, every consecutive feature map will be smaller after the convolution layer.

The space between two consecutive kernel places is named a stride, that describes the convolution operation as well. The usual choice of a stride is 1; so, a stride greater than 1 is occasionally applied so that to obtain down-sampling of the feature maps. Another option process to carry out down-sampling is a pooling operation, as defined below.

The convolutional operation has two significant features: local perception and parameter sharing. The main attribute of a convolution layer is weight sharing. Here, kernels are shared on the all the image locations. Weight sharing makes the features of convolution layers: decrease the number of parameters by increasing the model performance to learn in contrast with the FC layer (Rikiya Yamashita et al,. 2018).

3.2.2.2 Batch Normalization

To normalize the input layer by modifying and scaling the related activation, a normalization layer of cross-channel is applied. This generates a local response normalization layer with a window of a specific size centered on a channel-wise. Normalization can be applied in the acceleration of backpropagation and network training. Batch normalization operated for calculating the complexity of very deep model training and it is an adaptive re-parametrization procedure. Practically, it is very helpful to create it simpler to improvise deep neural networks. Basically, the deepest designs will be a combination of several operations. The training of the deepest networks is difficult, since the inputs to every layer is the parameters of all the layers are updated concurrently and, influenced by the parameters of whole previous layers. Once the parameters get updated, unanticipated outcomes could occur, for instance, gradient explosion and gradient vanishing, since several works or layers constructed with each other are updated concurrently. In this respect, it would be too hard to select a suitable learning rate such as stochastic gradient descent, since the influence of updating, the parameters of a layer robustly rely on the parameters of whole other layers.

Batch normalization offers an easy, influential path of reparametrizing close to any deep network. The reparametrizing of a deep neural network can importantly make an ease the issue of corresponding updates through several layers. Batch normalization could be applied to every input or hidden layers in a neural network (Szegedy, 2015).

3.2.2.3 Activation Function

In our models in this work we have used two different activation functions as:

3.2.2.3.1 Rectified Linear Unit (ReLU) Layer

The NN models apply a non-saturated activation function named ReLU. ReLU is obtained better performance and dramatically reduce the training time compared to other activation functions like a hyperbolic tangent or sigmoid activation function, as this function rectifies and avoids the vanishing gradient issue. The ReLU is a linear function that is followed by a convolutional layer, including feature maps.

The ReLU process is described by the following equation as a function of A in that the output is equal to the input once A is positive and 0 for other values. In Figure 3-7 the ReLU function is represented graphically.

$$f(A) = \max(0, \mathrm{A}) \qquad (3.1)$$

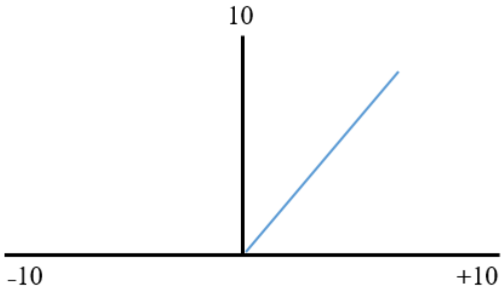


Figure 3-7**:** Graph of Rectified Linear Unit.

3.2.2.3.2 Softmax activation function

Typically, the activation function used the final FC dense layer in a variety way from all others. According to each duty, an effective activation function needs to be chosen. The activation function used for the multi-class classification operation, is a soft-max that normalizes the actual output values from the final fully connected dense layers to the possibilities of the target class, where every value ranges from zero to one and all values add up to one. Here, we have applied softmax as an activation function.

3.2.2.4 Maxpooling Layer

There are three phases of a standard convolutional neural network layer. First, multiple convolutions are performed by the convolutional layers to create a series of linear activations. Second, for each linear activation, a nonlinear activation function is implemented, such as the rectified linear activation function and the soft-max activation function. A pooling function will be used in the final phase to further change the nonlinear activations.

A pooling operation calculates a brief statistic of the closed outputs of the preceding layer at a definite position. For instance, the max-pooling process, as shown in Figure 3-8, takes the possible highest value within a rectangular neighborhood. The adequate of a rectangular neighborhood, the L2 norm of a rectangular neighborhood, etc… are other pooling functions. The insight of applying a pooling is that a feature's accurate position is less significant than its rough area. The solution of the feature descriptions, the amount of computation, and the number of parameters can be reduced by the pooling layer. It is useful for managing and avoiding of over-fit. A pooling layer among consecutive convolutional layers in deep CNN is commonly applied. The feature representations become nearly constant to a tiny translation by a pooling layer. Invariance to translation means that most of the pooled values do not modify if the input is translated by a tiny number. Invariance to local translation would be helpful if the exact location would be more significant when some feature is present.

As for the maximum pooling layer, it is an operation of down-sampling applied by separating the entire image into small rectangles (2×2) that pass over the image with a determined spatial in-variance and a way of (2×2). And then give priority to the four elements with a high possible value. The pooling layer is applied in decreasing the number of parameters and hence the number of network computations (Sultan et al., 2019).

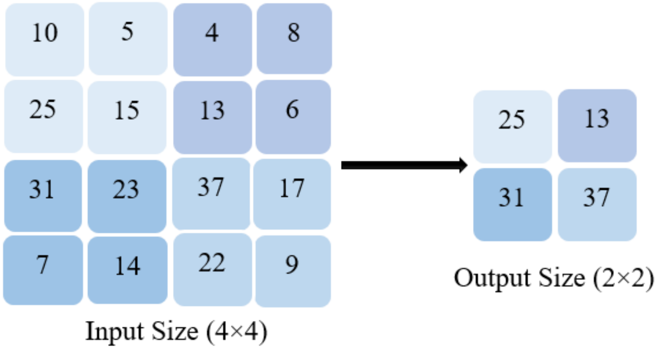


Figure 3-8: Maxpooling Layer Process.

3.2.2.5 Global Average Pooling

The global average pooling is consisting of another pooling function value noting. A strong method of down-sampling is carried out by a global average pooling, where a feature map with a height-to-width size is down-sampled into a 1-to-1 array by simply getting the average of all the elements in each feature map while maintaining the depth of feature maps. This process is usually operated merely one time before the FC layer. The profit of implementing average global pooling is to enable CNN to accept variable-size inputs and decrease the number of parameters that can be learned. It represented in Figure 3-9 (Rikiya Yamashita et al,. 2018).

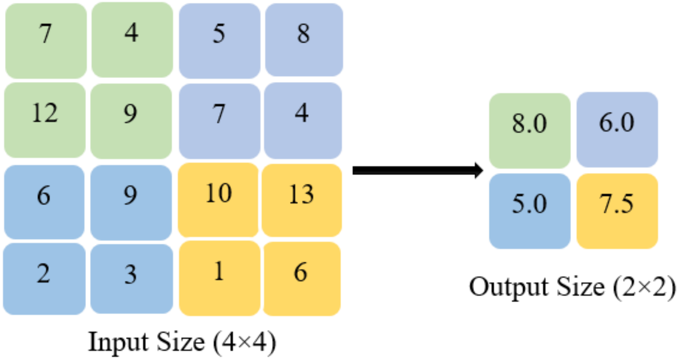


Figure 3-9: Average Pooling Layer Process.

3.2.2.6 Fully Connected Layer

The fully connected layers appear to be slower rather than the convolution operations because of a huge number of weights while every neuron in a FC layer is combined to each neuron in its previous layer. Once the fully connected layers get increase, the CNN appears inefficient and slow. In spite of the time restrictions, FC layers are significant field of CNNs, once responsibly they are determining the value for the significance of their contribution and individual features to the last output. Fully connected layers own a massive impact over output of the network, so applying a few number of elements in the fully connected layer increases the networks' precision and speed. (Sajid et al., 2019). The output feature maps of last convolution or pooling are normally flattened, for instance, connected to one or more FC layers, and transferred into a one-dimensional array of vectors, also called dense layers. Within that, by the learnable weight all the inputs have relation with all outputs. When the convolution layers extract the features, the pooling layers will product the down-sampling as well. Additionally, a subclass of fully connected layers map the aforementioned process to the final output of the network, as the possibilities to all class in the classification duties. The last FC layers normally have similar number of output nodes as the number of classes. A nonlinear function is following every FC layer, like ReLU, as it has been clarified previously (Rikiya Yamashita et al,. 2018).

3.2.2.7 Dense Layer

CNN starts with a dynasty of layers of convolution and pooling, and end with a layer that is fully connected. We can construct a CNN by stacking several convolutional, pooling, and FC (Dense Layer). We used the stack of alternative convolutional and pooling layers of the ResNet50 architecture. This section of the design is also known as the basis of convolution. The dense layer is a deep fully connected layer. We also introduced a new dense layer instead of the dense layer of the CNN and ResNet50 model that generates a multi-vector containing three classes. The annexed new dense layer is then trained on top of the basic convolutional. In addition, last layer of softmax has been introduced, which utilizes a feature of softmax activation and returns predictions instead of possibilities itself in the last layer. In the hidden layers, the ReLU activation function is used.

3.2.2.8 Dropout Layer

Over-fitting is always a problem that occurs because of the training data process and the massive number of weight parameters. In the training data, the network works well, but it products poor outcomes during the testing data. Regularization is applied to treat with the over-fitting issue that penalizes weights on layer-to-layer combinations. Dropout is a type of regularization, typically applied to deal with over-fitting. Dropout is a particular percent of neurons in a feature map that takes the dependence off among adjoin neurons, creating it generalized increasingly. It has been shown in Figure 3-10. (Srivastava et al,. 2014)

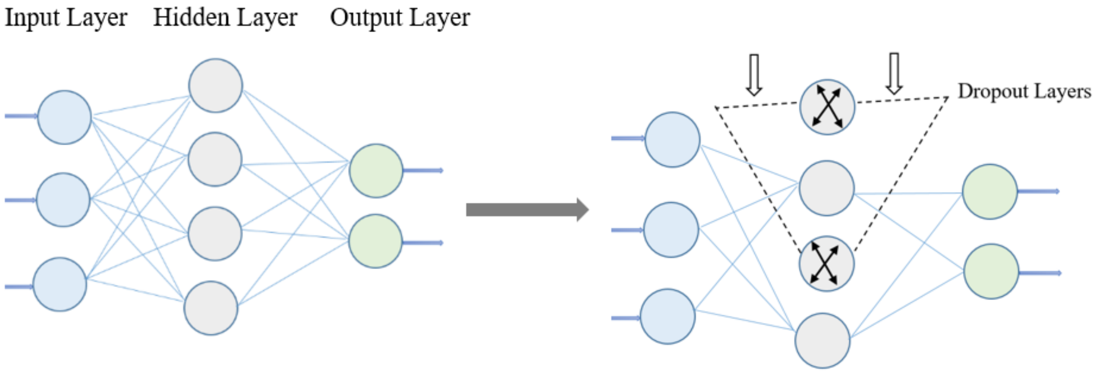


Figure 3-10**:** Dropout Layer Process.

3.2.2.9 Optimization

Optimization is applied primarily to reduce the loss function and update network parameters to attain the global minimum in the optimal task, by obtaining a small procedure to the negative gradient direction. In this work, we used Adam optimizer, which is a substitute algorithm for stochastic gradient descent for training deep learning methods. Adam combines the foremost elements of the Root Mean Square Propagation (RMSProp) and Adaptive Gradient (AdaGrad) algorithms to process an optimization algorithm that can manage dense gradients on noisy issues (Kurbiel & Khaleghian, 2017).

3.2.2.10 Loss Functions

A loss function is often known as a cost function, which tests the consistency among the network's output expectations by forwarded propagation and given ground truth labels. Cross-entropy is widely used loss function for multi-class classification. While, mean squared error is usually used for continuous values regression. One of the hyper-parameters is a kind of loss function and needs to be calculated according to the tasks given. (Mzoughi et al., 2020)

3.2.2.11 Training Network

Training a network is a procedure of discovering kernels on the training dataset in convolution and weights, also in FC dense layer that decrease variations among performance expectations and given ground-truth labels. The algorithm of back-propagation is the approach widely applied to train neural networks in which the gradient descent optimization and the algorithm of the loss function play important roles. A system efficiency under specific kernels and weights is computed by a loss function with forwarding propagation on training datasets, and learnable parameters, including kernels and weights, that are updated in an accordance with the loss value over an optimization algorithm named back-propagation and gradient descent.

3.2.2.12 Training Dataset

TL is an effective and common concept to train a network on a tiny dataset, where a network is pre-trained on a huge dataset, like ImageNet. That includes 1000 classes with 1.4 million images, then reused the function of interest provided. The main idea of TL is that public elements learned on a huge sufficient dataset could be allocated across apparently different datasets. This movability of learned public elements is a unique profit of DL which creates itself helpful in different domain functions along with limited datasets. Practically, there are two scenarios to employ a transfer learning pre-trained network: freezing and fine-tuning. In the below the both scenarios have been discussed in detail and how they worked. TL-based Pre-trained CNN model has been shown in Figure 3-11.

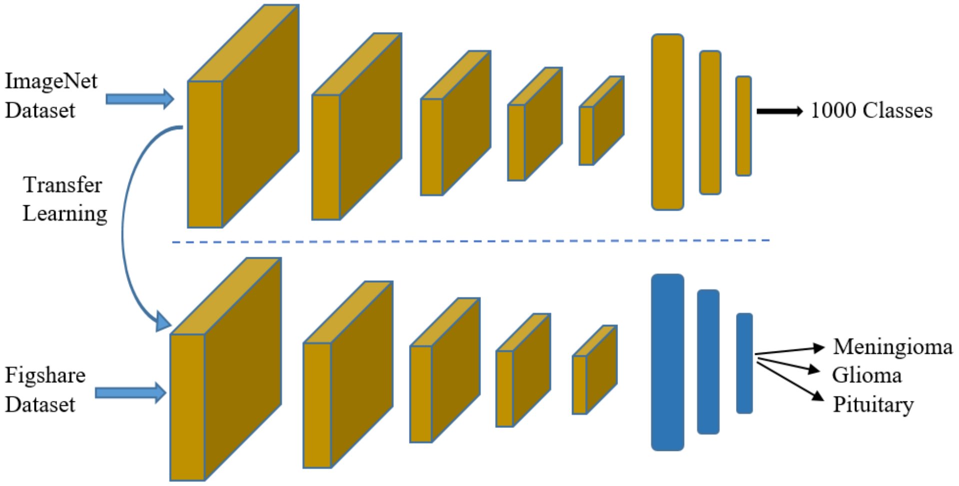


Figure 3-11**:** Is shown mechanism of TL, replaced the three final layers of the pre-trained CNN model.

3.2.3 Transfer Leaning

The goal of Transfer Learning is to enhance learning through the use of knowledge from the source tasks in the target tasks. Transfer learning is an efficient method for reducing the time needed for training. TL is a concept in which pre-trained techniques are used as a starting point for language processing and tasks in computer vision, as substantial computational. And time resources are needed to create neural network methods for these problems and because of the tremendous leaps in qualifications (Hassan et al., 2020). Transfer Learning techniques has become a very common process in image classification problems. In addition, for many datasets, off-the-shelf convolutional characteristics have shown a better classifier input than conventional hand-crafted image descriptors. These features are trained automatically by training convolutional neural networks on large-scale image datasets like ImageNet (Wacker et al., 2019). We often use transfer learning techniques in which a pre-trained CNN technique that is formerly modeled on a huge benchmark dataset such as ImageNet. It is applied in creating a scratched CNN technique for the image classification issue in deep learning. Rather than commencing the learning process from scratch, the prior learning is leveraged by transfer learning (Khan et al., 2020).

A powerful and innovative approach for Deep Learning is using Transfer Learning (TL) techniques to categorize Brain Tumors by extracting pivotal characteristics from a standard dataset. This is the major division of the presented work. To explore two distinctive Deep Learning models such as CNN and ResNet50 using Brain Tumor MRI images and apply Transfer Learning techniques to the given dataset. In the presented study, deep learning algorithms focused on Transfer Learning, that are assessed to reliably classify brain tumors into kinds of Meningioma, Glioma, and Pituitary. By using different architectures, the system is enhanced to achieve the best structure possible. It has shown as Figure 3-12. (Mehrotra et al., 2020).

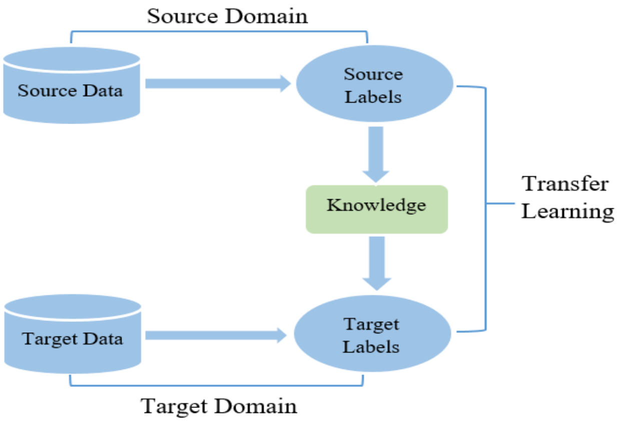


Figure 3-12**:** Transfer Learning block diagram architecture.

3.2.4 Support Vector Machine (SVM)

SVM is classified as the major efficient classification algorithm that can provide higher accuracy and performance compared to other classification algorithms. SVM is used for the classification of both linear and non-linear data types. This classifier is generated from the statistical learning that Vladimir Vapnik introduced in 1992. The SVM classifier determines the problem by finding the hyperplane with the largest margin, i.e. the highest marginal hyperplane. SVM has the particular property of increasing the geometric margin and simultaneously decreasing the classification error. For non-linear data, it is mapping the input vector into a higher-dimensional space, where a maximum hyperplane is built. It searches for linear optimal hyper-plane separation by translating it into high dimensional space with the aid of support vectors and margins (Mahalakshmi & Sumathi, 2020). Moreover, the binary classification is carried out by the Gaussian radial basis function (RBF). Many hyperplanes that increase the dividing margin between the classifications of brain tumor image types can be framed on the basis of the given training dataset. Furthermore, the extracted support vectors are shown at the hyperplane boundary line among classifications (Gokulalakshmi et al., 2020). In its simple definition SVM is used for binary classification. But in our work, we have used SVM for multi-class classification. In order to be able to use SVM for multi-class classification, we had to depend on the coding process and take advantage of the Scikit-Learn library in Python. But commonly there is another way to implement SVM that is using a mathematical formula to modify the structure of SVM in order to make it suitable for the multiclass classification. Particularly, in our thesis, we have implemented the support vector machine as a popular supervised ML model applied to the classification of three various types of brain tumors. By processing and analyzing the figshare dataset of MRI images, support vector machine classification supplies exact classification outcomes. We have used the kernel SVM here. In addition, the classification is executed with the creation of decision planes, by that the hyperplane separates the different class features. In particular, to spot the diagnosis of tumor existence from input brain images, the linear SVM-based classification algorithm is operated. Figure 3-13 shows the topology of SVM for multi-class classification.

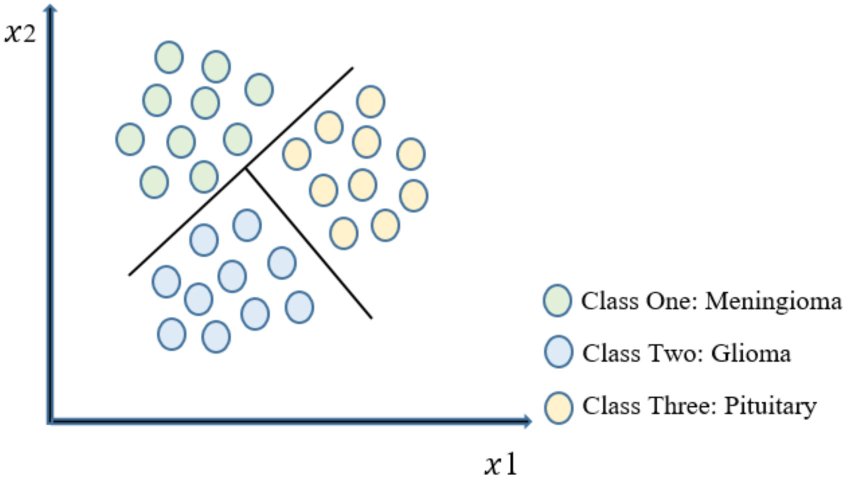


Figure 3-13**:** SVM for Multi-Class Classification.

CHAPTER 4

PROPOSED DEEP TRANSFER LEARNING FOR BRAIN TUMOR CLASSIFICATION

4.1 EXPERIMENTAL SETTING

The purpose of this thesis is to improve the accuracy of the classification of brain MRI images by applying DL methods, ML algorithm, and the approach of Transfer Learning (TL). TL is the task of using the knowledge given by a pre-trained framework to learn new systems provided by new data. It's typically simpler and much easier to calibrate a pre-trained system with TL rather than starting from basic. The use of pre-trained DL systems gives us ability to learn new works quickly. Here, we review the two distinctive DL models such as CNN and ResNet-50 using brain tumor classification MRI images and applying TL techniques to the given dataset. Pre-trained CNN and Resnet-50 models are used to execute TL to eliminate visually important and extraction of features. Finally, using the Support Vector Machine, the classification of these features is completed. It starts with the image dataset of figshare, which was collected and arranged into three distinct kinds of BT like Meningioma, Glioma, and Pituitary. The proposed strategy involves the following stages: first, preprocessing every MRI image, second, re-sampling in data augmentation technique, third, extraction of features based on DL and TL, finally the multi-class classification of various types of brain tumor by using SVM. Shown proposed system as Figure 4-1.

The key motivation for this thesis is to recognize and classify the tumor from MRI brain images in order to assist healthcare experts in properly treated patients. The suggested method included the following section in the successful tumor diagnosis phase.

1. Input figshare dataset
2. Preprocessing (Resize and Mask)
3. Data Augmentation (Re-sampling)
4. Feature Extraction
    - CNN with TL

- ResNet-50 with TL

5. Classification
    - Softmax
    - SVM

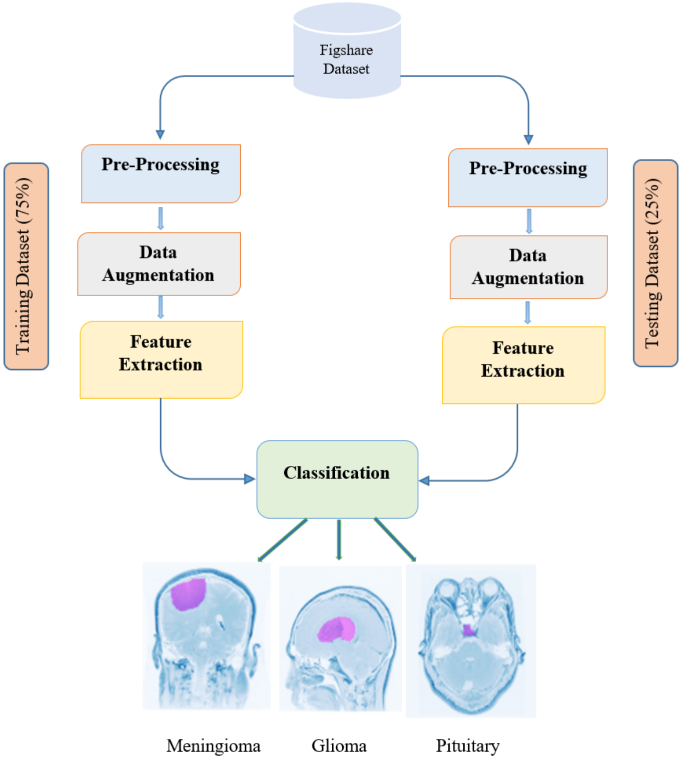


Figure 4-1**:** Diagram of the proposed system for the brain tumor classification.

4.2 DATASET DETAILS

The T1 weighted MRI brain tumor dataset is publicly open to the research community. The MRI image dataset includes 2-D images of three forms of brain tumors (meningioma, glioma, and pituitary). And the dataset includes three plane views of three kinds of brain tumors such as axial, coronal, and sagittal views as well. The dataset details are statically shown in Table 4-1. It includes 3064 MRI images of 233 patients from all three perspectives and different kinds of tumors. It also includes 708 brain MRI images of the meningioma corresponding to 82 patients, 1426 glioma images belonging to 89 patients, and the remaining 930 images refer to the pituitary tumor relating to 62 patients. The dimensions of each MRI images are 512x512 pixels (*Figshare brain tumor dataset*, 2018).

Table 4-1**:** Figshare Dataset Details.

Tumor type	**No. of patients**	**No. of images**	**MRI Views**
Meningioma	82	708	209 Axial 268 Coronal 231 Sagittal
Glioma	89	1426	494 Axial 437 Coronal 495 Sagittal
Pituitary	62	930	291 Axial 319 Coronal 320 Sagittal
Total	233	3064	

The most important details and differential information relating to any brain tumor are associated with the region’s position of the tumor along with its boundaries, size, and shape on any MRI image. The intensity, size, and shape of brain tumors have big deviations (Işin et al., 2016). Generally, meningioma is near the skull and cerebrospinal fluid. The shape of the glioma is distinct and generally it is encircled by edema. The well-known pituitary tumor is very near to the optic chiasma and sphenoidal sinus. It has shown in

Figure 4-2 (Noreen et al., 2020).

The classification of brain tumors is a more challenging research issue into sub-types. The following factors are due to the associated challenges in terms of intensity, size, and shape that the brain tumors show high variations. Tumors can show similar appearances from different pathological types. (Deepak & Ameer, 2019).

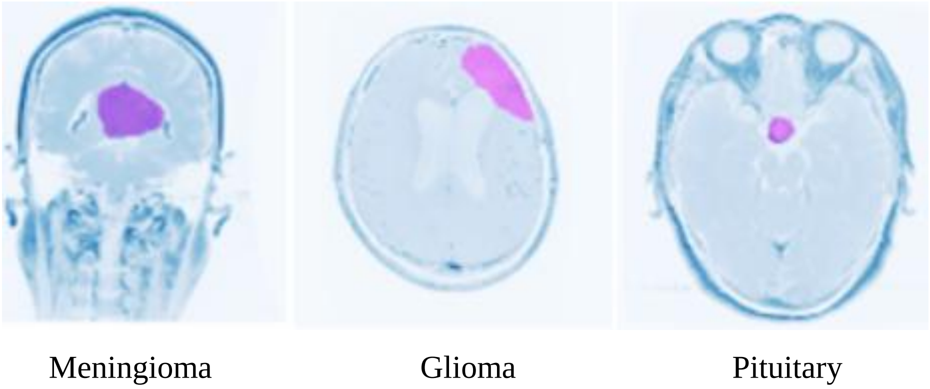


Figure 4-2**:** The location of brain tumor types. (1) Meningioma located near the skull. (2) Glioma including edema, necrosis, and surrounded by edema. (3) Pituitary located near the sphenoidal sinus.

4.3 PREPROCESSING

The first step in our thesis is preprocessing techniques. These techniques are very important for improving the quality of input images and providing suitable outcomes to help diagnose diseases in the processing of medical images. Cleaning the MRI images is the first step and activity of medical imaging analysis. It also helps to enrich the input image features, consisting of enlarging the rate of the signal-to-noise in the visual influence of the input samples. To improve the accuracy of the result, the pixel intensity of each input MRI image is obviously described. Moreover, the preprocessing techniques contain smoothing inner regions, unnecessary noise removal, and edge framing (Gokulalakshmi et al., 2020). In this study, as illustrated in Figure 4-3, our preprocessing consists of two steps: First, for memory optimization purposes, we resize the input MRI images of the entire set of 233 patients in the dataset that has 3064 MRI images totally. The size of the MRI images in the figshare database was 512 x 512 pixel. But we have resized the all images by reducing to 256 × 256 pixel, in order to make the work faster. Second, we utilized the mask method for diagnosis tumors at every MR image. It is used for designating the region, size, and shape of the tumors.

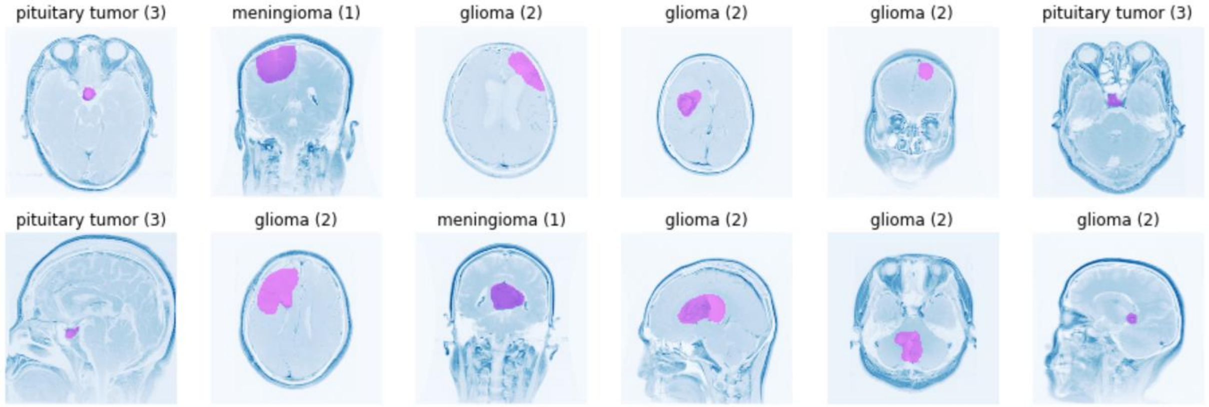


Figure 4-3**:** Resize and Mask for MRI images of brain tumor types.

4.4 DATA AUGMENTATION

The data augmentation in computer vision is a significant key reason which has a high influence on the training of deep learning models. Data Augmentation has different techniques like (flipping, scaling, rotation, crops, and shear). The profit of using this data's techniques is to increase the number of the datasets, and also decrease the overfitting problems during the deep learning models within the training process. In our work we apply the Data Augmentation to make three samples of each image. We re-scale the medical images to enter them into the network for the training process. The total number of Figshare MRI datasets contains 3064 images but after Data Augmentation the number became 9192 samples as the same size like the original image in the dataset that includes 256 pixels.

In this way, we achieve high accuracy and avoid overfitting issues. All the aforementioned process is called re-sampling that we used it as one of the data augmentation techniques. It has shown in Figure 4-4.

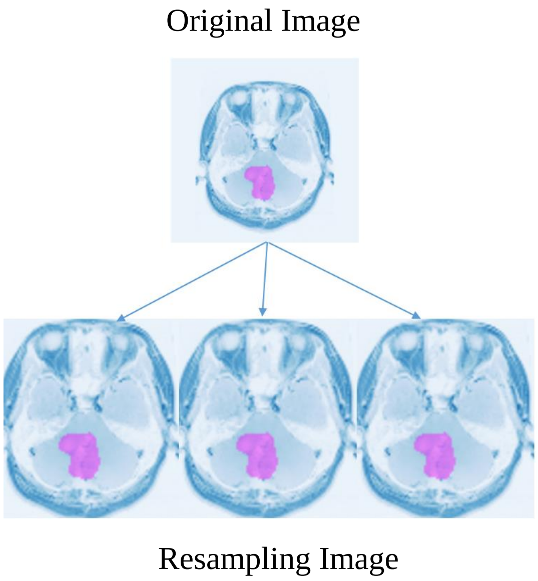


Figure 4-4**:** Re-sampling Process.

4.5 FEACTURE EXTRACTION

The second phase in our thesis is feature extraction steps. There are two issues here. First, it concentrates on either low-level or high-level characteristics only. The content of particular groups in the figshare dataset is divided with intrinsic irregularity. A powerful correlation exists between the edema, tumor layout, and normal tissues surrounding it. The shape of the meningioma and pituitary tumor is identical and these two different sorts of the tumor are usually not related to edema. The skull, gray matter, and cerebrospinal fluid are generally adjacent to meningioma. A pituitary is near the optic chiasma, internal carotid arteries, and sphenoidal sinus. Glioma appears to be dissimilar in form and is usually surrounded by edema. Second, the discriminative attributes and most important information of brain tumors are connected to the place of the tumor area in the MRI image together with its shape, size, boundary, and texture. For feature extraction, the key features of MRI scan images as size, shape texture, pixel intensity, and colors are considered. In our thesis two models of feature extractions have been used. They are convolutional neural network and ResNet-50; we have also used transfer learning for each of them.

4.5.1 Feature Extraction and Transfer Learning using the Proposed CNN Architecture

Several layers, containing convolution, pooling, and fully connected Dense Layer, are in the CNN structure. The phase in which input data via these layers are transforming into output named forward propagation. A similar process would be executed for 3D-CNN as well, while convolution and pooling operations are defined for 2D-CNN (Sultan et al., 2019). The suggested CNN architecture, beginning from the input layer from the prior preprocessing images phases pass over the convolution layers and their activation functions that applied in down-sampling and features extraction like convolution, normalization, ReLU, and pooling layers. To avoid over-fitting, a dropout layer is followed and applied by a fully connected dense layer and a softmax to anticipate the output and eventually a classification layer which generates the expected class. The CNN architecture that we have used for our thesis is consisting of different layers along with activation functions.

The proposed CNN model in this work that has been used for the feature extraction and classification of the figshare dataset MRI images are executed within the architecture that is consisting of (In the proposed CNN model the input layer's size is 256 × 256. The batch normalization in this work has been operated for calculating the complexity of very deep model training, and used for normalizing the input layer. Four convolution layers are employing the ReLU activation functions. Four max-pooling layers are provided after each

ReLU activation function. One dropout layer applied to avoid over-fitting. One flattened layer located behind these layers. Four dense layers (the three first dense layers that have to employ the ReLU activation and after the final dense layer used the softmax activation function). Lastly, Adam optimizer has been applied with categorical cross-entropy as the loss function). We have also used transfer learning along with CNN to avoid over-fitting problems. Meanwhile, we have combined the CNN and the transfer learning by saving the initial layers for feature extraction and fine-tuning (replacing) the last layers of the CNN model for classification. Besides all these, we have implemented SVM to diagnose the multi-class classifications of the types of brain tumors such as (meningioma, glioma, and pituitary). In Figure 4-5 the CNN architecture has been shown and explained.

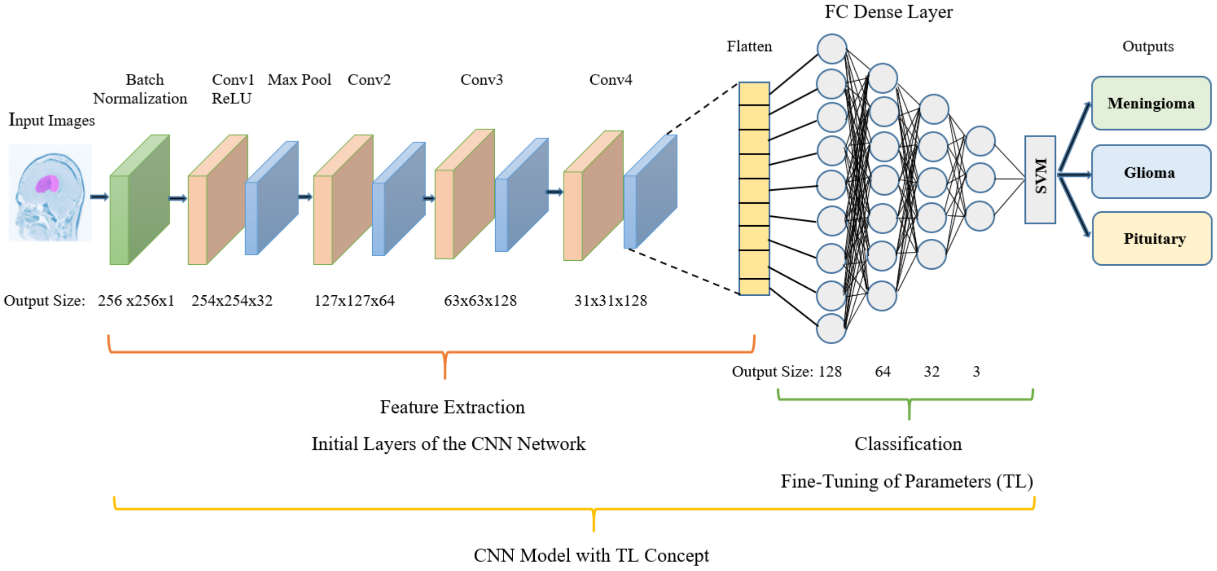


Figure 4-5: Building Block of CNN Architecture.

The below table is including (name of layers, out shapes, and number of parameters). And the total number of parameters:

- Total number parameters (3,869,639).
- Trainable parameters (3,869,637).
- Non-trainable parameters (2).

Table 4-2**:** CNN Architecture

Name of Layers	Output Shapes	Number of Parameters
Input Images	(None, 256, 256, 1)	0
Batch normalization	(None, 256, 256, 1)	4
Convolution	(None, 254, 254, 32)	320
Max-pooling	(None, 127, 127, 32)	0
Convolution	(None, 127, 127, 64)	32832
Max-pooling	(None, 63, 63, 64)	0
Convolution	(None, 63, 63, 128)	73856
Max-pooling	(None, 31, 31, 128)	0
Convolution	(None, 31, 31, 128)	65664
Max-pooling	(None, 15, 15, 128)	0
Dropout	(None, 15, 15, 128)	0
Flatten	(None, 28800)	0
Dense	(None, 128)	3686528
Dense	(None, 64)	8256
Dense	(None, 32)	2080
Dense	(None, 3)	99
Total parameters:		3,869,639
Trainable parameters:		3,869,637
Non-trainable parameters:		2

4.5.2 Pre-trained ResNet-50 Model

ResNet is a DCNN model, introduced by He et al. in Microsoft at 2015. The ResNet took a prior position in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a 3.57% error rate (He et al, 2015). In a residual network, instead of learning features, we learn from residuals that deduct attributes learned from the inputs of the layer. The skip connection was used by ResNet to relay information over layers. ResNet directly connects the nth layer input to some (n+x) layer, allowing additional layers to be stacked and a deep network to be created. In our experiment, we used a pre-trained ResNet50 model and Transfer learning (Khan et al., 2020). There are several versions of the same conception of ResNet architecture, but with a various number of layers such as Resnet18, Resnet34, Resnet50, Resnet101, Resnet110, Resnet152, Resnet164, and Resnet1202.

When opposed to VGG with up to 16 and 19 layers and AlexNet with 8 layers, the ResNet structure popularized the concept of using deeper networks. Skip implemented by the ResNet architecture connections, often referred to as residual connections, prevent the loss of information while deep network training operated. The skip connection process allows very deep networks to be trained and can improve the performance of the model. The structure of ResNet is primarily composed of residual blocks. Successive hidden layers are connected to each other in shallow neural networks, but there are links among residual blocks in the ResNet structure as well.

The most persuasive profit of the residual network in the design of ResNet; the connections maintain the knowledge obtained during training and speed up the model's training time by increasing the capacity of the network (Talo, 2019). In this work, a block diagram of the pre-trained Resnet50 technique applied. ResNet50 used the idea of residual learning for higher performance and better parameter optimization. The residual learning idea is illustrated in Figure 4-6. When the activation function F(x) returns zero, it detours the identification of the mapping block y=x, where x shows the layer input. The residual learning method decreasing the over-fit and provides classification models that are generalized.

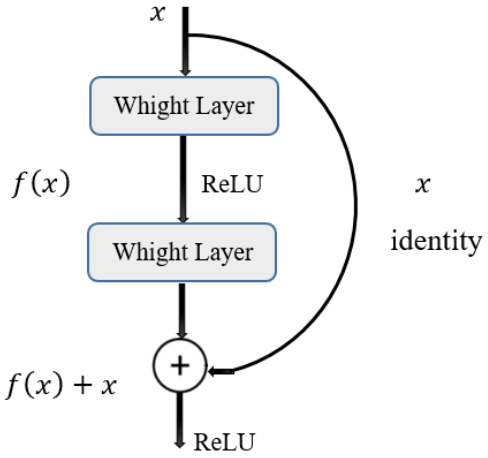


Figure 4-6**:** Building Block for Residual Learning.

4.5.2.1 Feature Extraction and Transfer learning using ResNet50

First, we carry out the pre-processing phase to resize and mask of figshare MRI dataset and match it with ResNet50 input size. In the TL-ResNet50 training model, we achieved the ResNet50 technique and used the transfer learning approach on the final fully-connected layers (FC, softmax, and classification). The ResNet50 was originally trained on 1,000 classes of ImageNet datasets (Russakovsky et al., 2015).

The pre-trained ImageNet weights exclude the final FC layers of ResNet50, they are frozen and used for creating a system to the mood of classification issue. It is suitable for applying the transfer learning model on the pre-trained DL method on the moderately low dataset, rather than training the techniques of scratch on, that needs a huge dataset. Hence, we trained the network to learn FC weights for the classification of brain tumor modalities. This converted TL-ResNet50 network has here been trained and fine-tuned rather than 1,000 on a new dataset of 3 classes. In extract deep feature process, TL-ResNet50 model is trained on figshare database to deep feature extraction on the "average-pool" layer prior final FC layers. The TL-ResNet50 functions as an undetermined feature extractor, it authorizes the new input image to stop and forward propagation at a predefined (avg-pool) layer to achieve deep features. By freezing the pre-trained ImageNet weights, we also can leverage the discriminating and robustness learning ability of TL-ResNet50. An optimal deep feature vector of size 2048 has been achieved at the (avg-pool) by hiring the transfer learning model for classification state. Deep features of the high classification accuracy achieved are fed to SVM for final classification (Hassan et al., 2020).

ResNet50 is a residual network with 50 layers. The ResNet-50 model includes five convolution steps. In the first step Conv1 consisting of only one convolution layer, and it has merely one convolution block as well. The remaining layers consist of (Conv2 as it contains three convolution blocks, Conv3 includes four convolution blocks, Conv4 consists of six convolution blocks, and three convolution blocks are forming Conv5). Every three layers of Conv (1×1), Conv (3×3), and Conv (1×1) belong to a convolution block.

The Average pooling layer with the Down-sampling is shifting the size of the feature map. Except those, an FC layer is existing for classification aim from the conclusion of the model. In this thesis, we have used ResNet-50 because it is a huge and powerful network for classification in medical image fields. We have specifically used it for the classification of brain tumor types. ResNet-50 is generally used for a giant dataset, but here in our thesis, we have used it to classify the brain tumor types in Figshare MRI image datasets which these datasets are smaller. In our work, we have used transfer learning along with ResNet-50 to avoid error, over-fitting, and missing data. The main profit of using these two techniques together in our thesis has mostly raised the performance of the classification of brain tumors. We have also functioned both techniques practically by saving the initial layers for feature extraction and fine-tuning (replacing) the final layers for classification. Good results have been achieved through that practical operation, but also we have used SVM as an algorithm for multi-class classifications and it has a very good performance, accuracy, and very good outcome. As ResNet50 method illustrated in Figure 4-7.

In this work:

- Total number parameters (23,587,523).
- Trainable parameters (23,534,403).
- Non-trainable parameters (53,120).

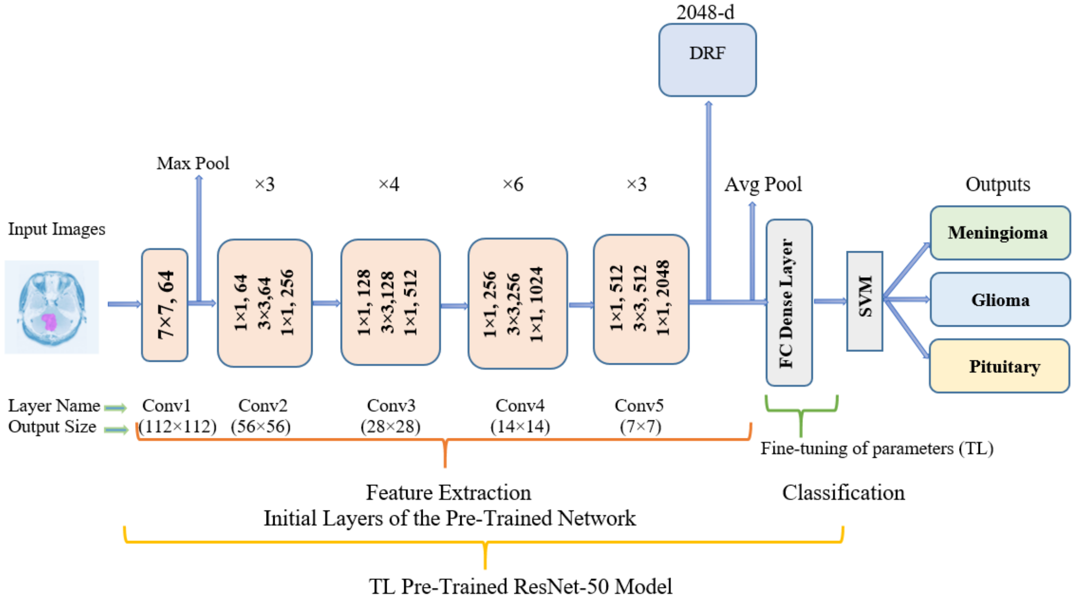


Figure 4-7**:** ResNet-50 Diagram Architecture.

4.5.3 Transfer Learning Scenarios

In commonly, two significant transfer learning scenarios are presented: **fine-tuning** and **freezing**.

4.5.3.1 Fine-tuning: Biases and weights of a pre-trained CNN are implemented, instead of random initialization. And then a conventional training procedure on the target dataset is conducted. The fine-tuning of TL is applied by replacing the last layers of the pre-trained network to enhance the performance and effectiveness of the convolutional neural network. In this case, instead of retraining and replacing the whole design of the CNN classifier, ConvNet weights are initialized from the top of the CNN and ResNet50 pre-trained network (PTN). This idea functions by moving weights from the ImageNet (source dataset) to our Figshare (target dataset) for the CNN and ResNet50. The standard operation is to shorten the softmax layer of the PTN and replace it with our new softmax layer that is related to our issue. In this thesis, applied every structure of CNN. The final FC dense layer is replaced with many of the target datasets. That is to say, 1000 classes of ImageNet are replaced with the three classes of figshare brain tumor dataset. It has shown as Figure 4-8.

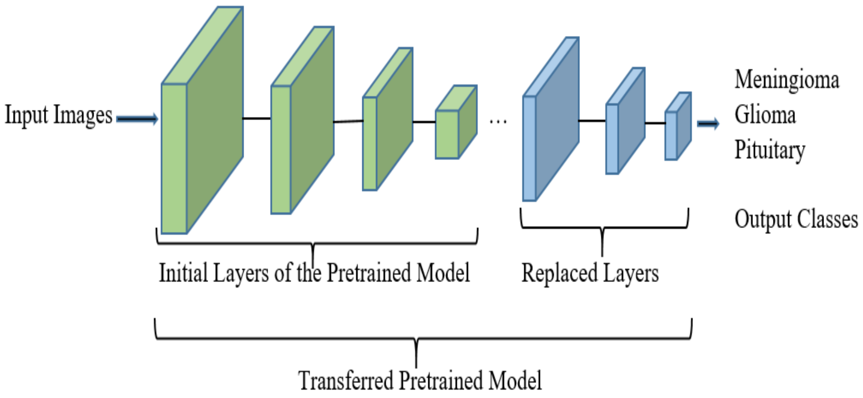

Input Images
Meningioma
Glioma
Pituitary
Output Classes
Initial Layers of the Pretrained Model
Replaced Layers
Transferred Pretrained Model

Figure 4-8: Transfer Learning concept using pre-trained scenario.

4.5.3.2 Freezing scenario: We consider the pre-trained CNN layers as constant feature extraction. We freeze the biases and weights of our required convolutional layers in this context and allow the fully connected layers to be fine-tuned over the target dataset. In this model, pre-trained network layers are worked and frozen as constant elements. This idea functions with concluding the weights from the ImageNet (source dataset) of the pre-trained model. And the arbitrary vector features can be applied from convolutional layers or from fully connected to train a linear (SVM) classifier on the Figshare (target dataset).

4.6 CLASSIFICATION

The final phase in our thesis is classification. For humans, the classification between images is a simple task, but it has proved to be challengeable for machines. Higher-capacity computing and high-end and cheap video cameras, along with the increasing need for automated video analysis, have generated an interest in algorithms for object classification. A simple grading model include a camera placed high above the region where images are registered and thus processed. Preprocessing, feature extraction, image sensors, object detection with division, and object classification are included in the classification. The classification systems are databases including predefined patterns which compared to detected objects. They should be classified into the correct category. In various fields of use as biometrics, biomedical imaging, remote sensing, vehicle navigation, widespread surveillance, robot navigation, industrial visual inspection and image classification are significant and challenging roles (Mahalakshmi & Sumathi, 2020). Support Vector Machine (SVM) is the most common technique in the classification of brain tumors. It is generalization capacity and for overcoming the limitations of classifying the non-separable type of data (Mohsen et al., 2017). In this work, the last implementation is the multi-class classification based on the features that is extracted from MRI images of the three various kinds of brain tumors into meningioma, glioma, and pituitary. In the field of classifying brain MRI images, supervised classification techniques had better performance where the dataset is randomly divided into two processes: one of them is to train the classifier, and the other is to test and evaluate the classifier.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Performance evaluation

Transfer Learning concept is assessed in this thesis to the accurate classification of brain tumor sorts as meningioma, glioma, and pituitary. The concept is trained by applying various Deep Learning models including CNN and ResNet50 pre-trained networks to achieve the better accuracy of the system. The image multi-class classification is done by using SVM. Thus, the trained visually discernible attributes from each DL network are modified to the objective dataset and the SVM performs the Brain Tumor classification by installing the number of neurons in the figshare dataset to three classes.

5.2 Experimental setup

We have evaluated the setting of our results by depending on the experiment that is done by applying Keras, TensorFlow, and Scikit-Learn libraries in python for deep learning and machine learning methods. For implementation, we used the Jupyter tool in Anaconda software. Figshare dataset for brain tumor types was used. The data split training and testing, as 75% of the dataset was applied for training and 25% for testing. We have divided our work into two parts: The first part, which consists of CNN with Transfer Learning (TL) as well as SVM for the Figshare dataset. The second part includes Resnet-50 with TL, as well as SVM for the same dataset. Both implementations have good results, all the outcomes are remarked in this chapter.

5.3 Evaluations and Results

In this section, we compare the classification results of CNN with transfer learning (CNN+TL), CNN+TL with SVM classifier, Resnet-50 with transfer learning (Resnet-50+TL), Resnet-50+TL with SVM classifier. In this way, we illustrate that using an SVM classifier with transfer learning improves the performance significantly. Finally, we compare our results with other state-of-the-art methods on the Figshare dataset and show that our approach is effective. We split the dataset to training and testing, so we use 75% of the data for the training and 25% for the testing, and then separate the testing set to 50% validation and 50% testing. It has shown in Figure 5-1 and Figure 5-2.

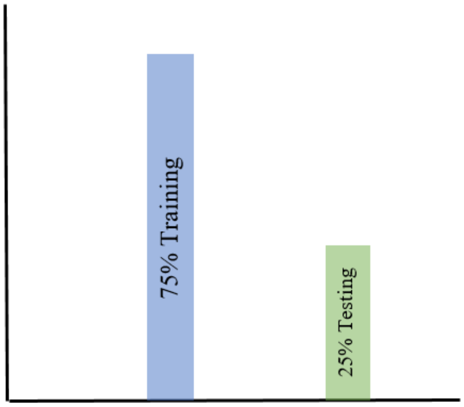


Figure 5-1**:** 75% for training and 25% for testing in the figshare dataset.

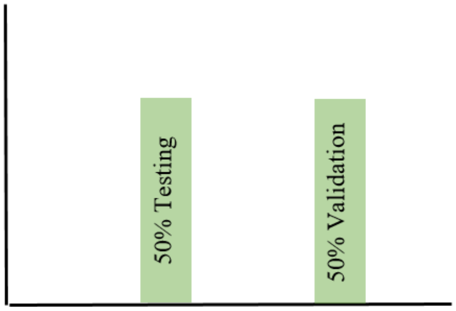


Figure 5-2**:** 25% Testing dataset (50% for testing and 50% for validation)

5.4 Classification accuracy prediction

There are different methods to assess a model for classification:

5.4.1 Accuracy

We used the qualified networks in this section to classify the test images and
the measurement of all classification accuracy. The accuracy of classification is the capability to correctly predict and guess the value of new data for a predicted attribute. Accuracy is the main performance measurement metric for classification. It is the number of accurate predictions divided and multiplied by 100 by the total number of predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+PN} \times 100 \qquad (5.1)$$

Where TP and TN are results generated when the system accurately classifies the positive class and the negative class, sequentially. Although FP and FN are results generated when the system inaccurately classifies the positive class and the negative class, sequentially.

5.4.2 Precision

If there are imbalanced observation points in the data set, then the classification accuracy is not a good model performance measure. In this state, even though you predicted all instances as the highest class, you will get a high rate of accuracy that makes no logic at all. Hence, the model does not know something, so it only predicts anything like the highest level. Therefore, for validation, class-specific performance metrics should be required. Precision is one of such metrics, which is described as:

$$Precision = \frac{TP}{TP+FP} \qquad (5.2)$$

For each class of the model, the above mathematical formula applied and validate the performance. If the precisions of all classes are just about the same, then it can be deduced that the system has trained for all classes equally.

5.4.3 Recall

It is another significant metric, described as the fragment of observation points from a category that is successfully expected by the model.

$$Recall = \frac{TP}{TP+FN} \qquad (5.3)$$

5.4.4 F1-Score

It is another significant measure to combine precision and recall in a single metric. It is the harmony means of precision and recall. F1-Score's mathematical equation meaning is described as:

$$F1 - Score = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{Precision+Recall} \qquad (5.4)$$

5.5 Performance Measures

In this section, the three types of brain tumor are classified, and we get results for each class separately, and also expect the accuracy we use the other measures for the performance like precision, recall, and f1-score. Later on, we have the average measures of precision, recall, and f1-score for each class (Meningioma, Glioma, and Pituitary tumor). As it has been shown in Table 5-1 that we have used CNN with TL, in Table 5-2 that we have applied CNN, TL, and

SVM, in Table 5-3 we have used Resnet-50 and Tl, and also in Table 5-4 we have applied ResNet-50, TL and SVM.

Table 5-1**:** The precision, recall, F1-Score based on CNN with TL for types of brain tumor classes.

Classes	precision%	recall%	f1-score%
Meningioma	0.9945	1.0000	0.9972
Glioma	1.0000	0.9972	0.9986
Pituitary tumor	0.9955	0.9955	0.9955
Average	0.9967	0.9976	0.9971

Table 5-2**:** The precision, recall, F1-Score based on CNN with TL and SVM for types of brain tumor classes.

Classes	precision%	recall%	f1-score%
Meningioma	1.0000	0.9833	0.9916
Glioma	1.0000	0.9945	0.9972
Pituitary tumor	0.9782	1.0000	0.9890
Average	0.9927	0.9926	0.9926

Table 5-3: The precision, recall, F1-Score based on ResNet-50 with TL for types of brain tumor classes.

Classes	precision%	recall%	f1-score%
Meningioma	0.9882	0.8789	0.9304
Glioma	0.9912	0.9628	0.9767
Pituitary tumor	0.8721	0.9912	0.9278
Average	0.9505	0.9443	0.9450

Table 5-4: The precision, recall, F1-Score based on ResNet-50 with TL and SVM for types of brain tumor classes.

Classes	precision%	recall%	f1-score%
Meningioma	1.0000	0.7000	0.8235
Glioma	1.0000	0.9083	0.9520
Pituitary tumor	0.7184	1.0000	0.8361
Average	0.9061	0.8694	0.8705

We have used CNN and Transfer Learning for the model accuracy in order to obtain training accuracy and validation accuracy, while our training model has reached 30 epochs. As has been shown in Figure 5-3. In Figure 5-4 we have achieved the training loss and validation loss while we have used CNN and TL, meanwhile, the number of epochs is 30 as well.

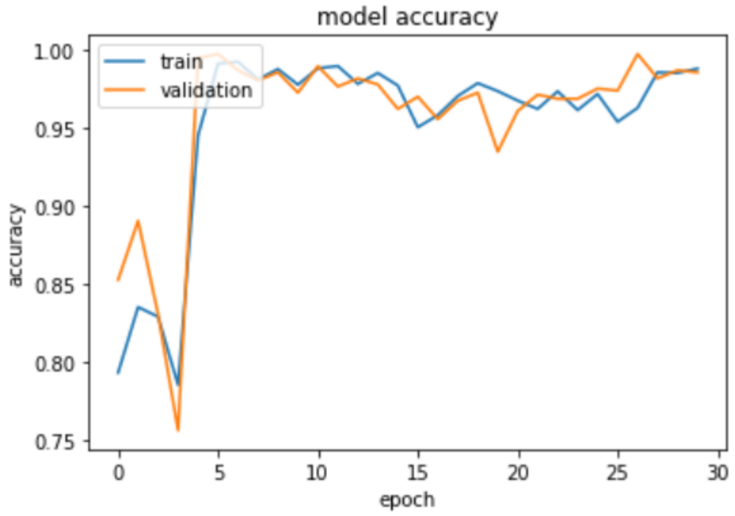


Figure 5-3**:** Train Accuracy Vs Validation Accuracy in using CNN with TL.

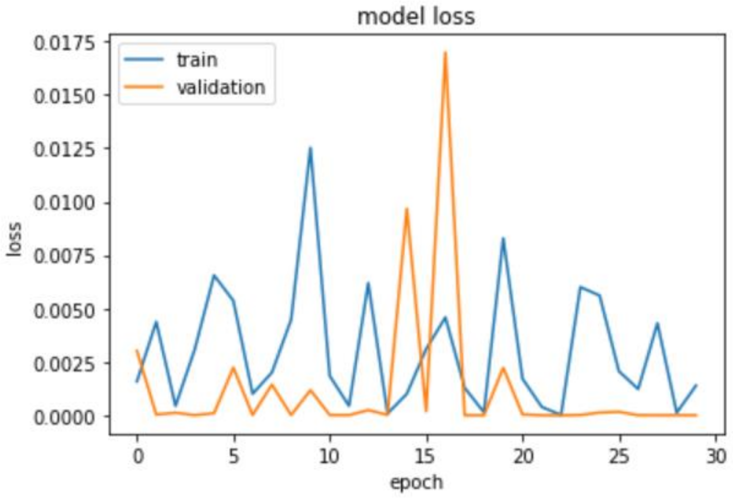


Figure 5-4**:** Train Loss Vs Validation Loss in using CNN with TL.

We have used ResNet-50 and Transfer Learning for the model accuracy in order to obtain training accuracy and validation accuracy, while our training model has reached 30 epochs. As has been shown in Figure 5-5. In Figure 5-6 we have achieved the training loss and validation loss while we have used ResNet-50 and TL, meanwhile, the number of epochs is 30 as well.

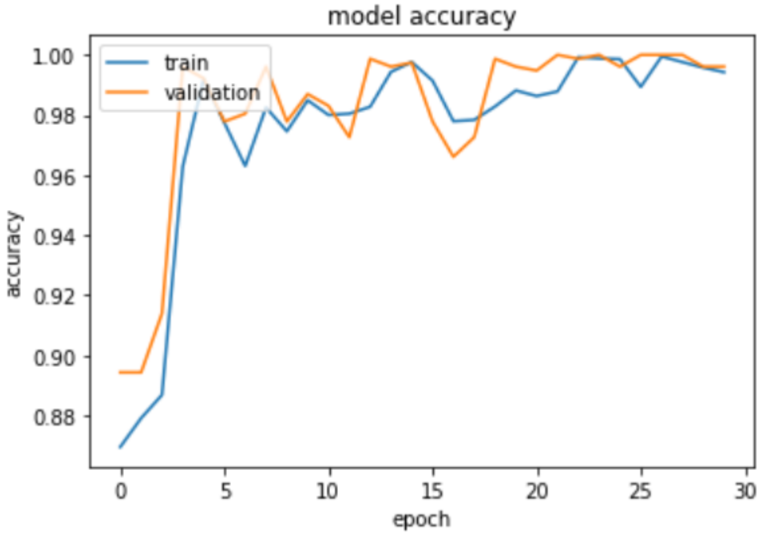


Figure 5-5**:** Train Accuracy Vs Validation Accuracy in using ResNet-50 with TL.

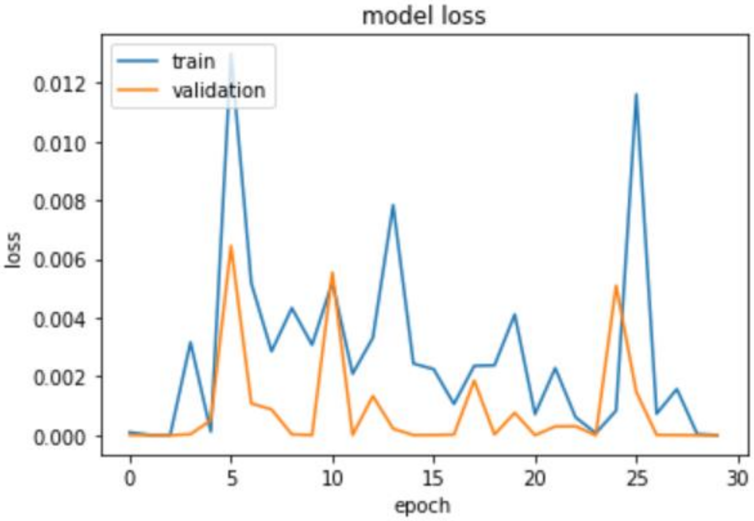


Figure 5-6**:** Train Loss Vs Validation Loss in using ResNet-50 with TL.

5.6 The Achieved Accuracy of the two Models

Accuracy classification is the most important performance metric for a category, which provides a percentage of the classifier's true predictions. We present the accuracy of the classification achieved in four different settings in the experiment. As illustrated in Table 5-5.

- When both models are used together as a standalone deep learning classifier, the accuracy of the modeled CNN with TL is 98.56%.
- The accuracy of the built CNN with TL achieved an accuracy of 99.35% with a feature of the SVM classifier.
- When both models are used together as a standalone deep learning classifier, the accuracy of the modeled pre-trained ResNet50 with TL is 99.61%.
- The accuracy of the pre-trained ResNet50 designed with TL, with a feature of the SVM classifier, achieved an accuracy of 88.38%.

Table 5-5**:** The Achieved Accuracy of the two Models.

Model Name	Model Accuracy (%)
CNN+TL	98.56
CNN+TL+SVM	99.35
Resnet-50+TL	99.61
Resnet-50+TL+SVM	88.38

5.7 Confusion Matrix

In a standard data classification issue, the evaluation metric has been used in two steps, which are the testing step and the training step. Meantime, in the testing step, the evaluation metric was employed as the evaluator to calculate the success of manufactured categorize when tested with the hidden data. In the training step, the evaluation metric was employed to optimize the algorithm of classification. In other words, the evaluation metric was used as the discriminator to discriminate and to choose the optimal solution which can manufacture a more precise prediction of future evaluation of a specific classifier.

The confusion matrix is applied here to verify performance. This matrix provides useful information about the predicted and actual labels given by the classification of the proposed model. Using this information, the results are calculated from various elements. A confusion matrix displays the synopsis of expectations made by the method, where each column shows the true label and each row shows a predicted label. A normalized confusion matrix is the outcome of the values split by the number of characteristics in each label for an improvement optic explanation of which label is being led to classify improperly. The confusion matrix is applied to evaluate the performance of model classifications, in which the predicted label is the number of the classes. In this study, we have three classes. Confusion matrix outcomes are shown in Figure 5-7 and Figure 5-8 for True-label and Predicted-label data distributions sequentially. Also, as illustrated in Table 5-6 and Table 5-7.

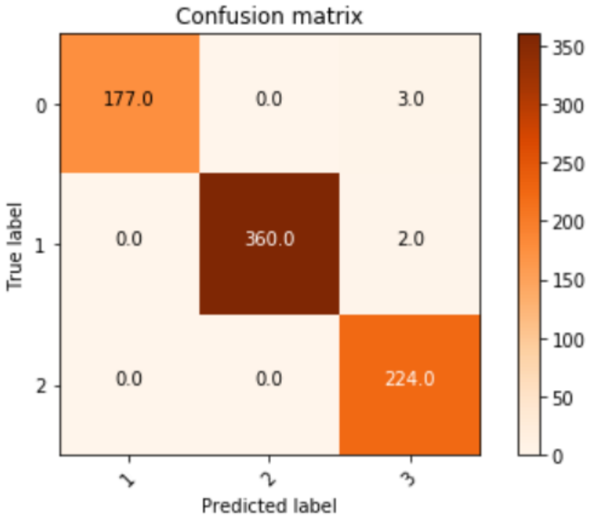


Figure 5-7**:** Confusion Matrix in using CNN with TL and SVM.

Table 5-6**:** Confusion Matrix in using CNN with TL and SVM for Classification of Brain Tumor Types.

	Meningioma	Glioma	Pituitary
Meningioma	177	0	3
Glioma	0	360	2
Pituitary	0	0	224

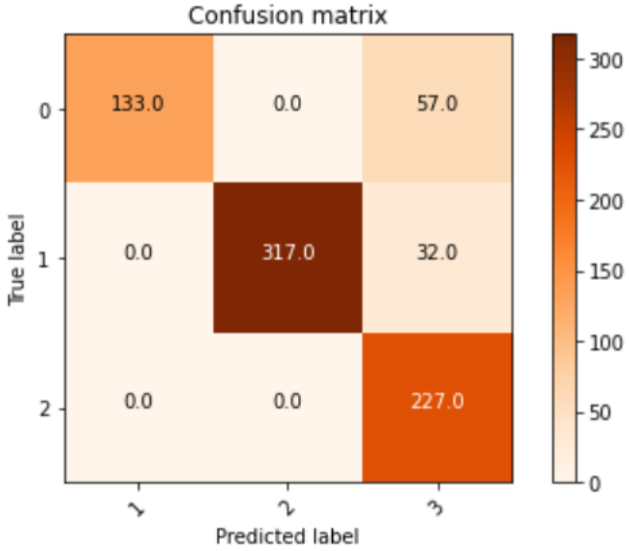


Figure 5-8**:** Confusion Matrix in using ResNet-50 with TL and SVM.

Table 5-7**:** Confusion Matrix in using ResNet-50 with TL and SVM for Classification of Brain Tumor Types.

	Meningioma	Glioma	Pituitary
Meningioma	133	0	57
Glioma	0	317	32
Pituitary	0	0	227

5.8 Comparisons and related works using figshare dataset

The suggested method is considering to be similar to the whole work in the particular issue of the classifying process. For two factors, classification accuracy is operated as a standard for comparing. First, in all the relevant works, the classification is the useable standard metric. Second, using the same dataset, all the associated researches is estimated. Table 5-8 describes the works and provides a comparison of results. SVM was an important approach to the BoW feature set design among the studies that used up hand-crafted models (Cheng et al. 2015). The outcomes have improved with the acceptance of DL methods and the operation of CNN elements (Pashaei et al. 2018; Swati et al. 2019). However, in accordance with our studies, when CNN, pre-trained ResNet50 with TL elements are categorized functioning a Support Vector Machine technique, the outcomes have improved further. The highest accuracy was obtained by our methodology, which applies a combination of attributes from the structured CNN and multi-class SVM for classifying. Our work has four different results. Initially, we use four dense layers in CNN, the first three dense layers are followed by the ReLU activation function. The last dense layer is followed by the softmax activation function. We use all these for the classification and achieve the result of 98.56%. In the next step, we use SVM with CNN and TL for the multi-class classification. We achieve the result of 99.35%. After that, we use the dense layer that is followed by the softmax activation function in ResNet-50 along with TL for the classification. We obtain an accuracy of 99.61%. Actually, the ResNet-50 is a powerful model that is used merely for classification. But we use SVM besides the ResNet-50 and fine-tuning that causes the rise in the accuracy of the classification. And also the achieved result is 88.38%. In the training of our work, we use 30 epochs for each CNN with the TL and the ResNet-50 with TL. In this way, we can understand that using the CNN which we have created shows that it is faster than the ResNet-50 within the training operation. And also it gives the best result. We have implemented run on our work three times, for each time the models repeatedly improved their learning by obtaining better results frequently.

Table 5-8**:** Comparisons and related works using figshare dataset.

Authors	Features	Model Classifier	Dataset	Accuracy (%)
Cheng et al. (2015)	BoW	SVM	Figshare	91.28
Ismail & Qader. (2018)	WDT & Gabor	BPNN	Figshare	91.90
Afshar et al. (2018)	CapsNet	CapsNet	Figshare	86.56
Pashaei et al. (2018)	CNN	ELM	Figshare	93.68
Abiwinanda et al. (2018)	CNN	CNN	Figshare	84.19
Anaraki et al. (2019)	CNN-GA	CNN-GA	Figshare	94.20
Liu et al. (2019)	ResNet-34	Gap	Figshare	95.00
Swati et al. (2019)	VGG19 (TL)	VGG19(TL)	Figshare	94.82
Ghosal et al. (2019)	ResNet-101	CNN	Figshare	93.83
Deepak et al. (2019)	**CNN**	**KNN**	**Figshare**	**98.00**
Deepak et al. (2020)	CNN	SVM	Figshare	95.82
Togacar et al. (2020)	CNN	CNN	Figshare	96.05
Rehman et al. (2020)	**Fine-tune-VGG16**	**Softmax**	**Figshare**	**98.69**
Proposed Method	**Fine-tune-CNN**	**Softmax**	**Figshare**	**98.56**
Proposed Method	**Fine-tune-CNN**	**SVM**	**Figshare**	**99.35**
Proposed Method	**Fine-tuneResNet-50**	**Softmax**	**Figshare**	**99.61**
Proposed Method	**Fine-tune-ResNet-50 (TL)**	**SVM**	**Figshare**	**88.38**

CONCLUSION

A brain tumor is a more destructive illness, heading to the lowest survival span at the largest degree. Any wrong diagnosis of tumors on the brain makes misunderstanding medical intervention and decreases patients' chances of survivability. The specific detection of brain tumors is a crucial point for adequate care planning to cure patients with brain tumor disease and improve their existence. We have suggested deep learning techniques for the classification of brain tumors from MRI images. Our method has been trained on a Figshare dataset consists of 3064 images of benchmark brain tumor MRI images. We are comparing two studies using convolutional neural network architectures and ResNet-50 pre-trained for feature extraction and in both parts of this thesis, we have applied transfer learning techniques into fine-tuning and for each one separately. Furthermore, we have functioned SVM for the classification of various sorts of brain tumors into meningioma, glioma, and pituitary in the proposed method. The aim of this work is to make the accuracy of the classification better, prevent overfitting and speed up the training time. In this thesis, in accordance with research on its effect on classification efficiency and time consumption, we trained our construction with minimum preprocessing for various epoch numbers. Additionally, with a few epochs in a limited time, the study obtained appropriate outcomes. In this work the highest classification accuracy of 99.61% obtained.

In our future research:

- We want to apply other data augmentation processes like (Rotating, Filliping, Cropping, and Random Erasing, … etc.) for increasing the number of datasets, and it is a useful technique for the reduction of overfitting problem.
- However, we want to work on the other types of pre-trained models such as AlexNet, GoogleNet, DenseNet, and other sorts of ResNet or VGG for detection, segmentation, and classification for the Figshare dataset and BraTS dataset. Brain Tumor Segmentation (BraTS) MRI is a dataset in which any subject consists of T1, T2, T2w, T1 Gd, FLAIR, and segmentation maps. There are two types of brain tumors malignant and benign. Glioma is a malignant primary brain tumor. It includes High-Grade

Glioma (HGG) and Low-Grade Glioma (LGG). This dataset includes a total of 396 subjects: 320 subjects with HGG and 76 subjects with LGG.

- Presently, segmentation is one of the most significant tasks in the area of computer vision, especially in the medical image field. Segmentation of the brain tumor is a vital procedure and needs a group of clinical experts to accurately describe the shape, size, location, and intensity of the tumor. Also, we want to use different sorts of pre-trained deep learning methods for the detection, segmentation, and classification of various types of brain tumors in the Figshare dataset and BraTS dataset.

REFERENCES

Gumaei, A., Hassan, M. M., Hassan, M. R., Alelaiwi, A., & Fortino, G. (2019). A Hybrid Feature Extraction Method with Regularized Extreme Learning Machine for Brain Tumor Classification. *IEEE Access*, *7*, 36266–36273. https://doi.org/10.1109/ACCESS.2019.2904145

Sultan, H. H., Salem, N. M., & Al-Atabany, W. (2019). Multi-Classification of Brain Tumor Images Using Deep Neural Network. *IEEE Access*, *7*, 69215–69225. https://doi.org/10.1109/ACCESS.2019.2919122

Rehman, A., Naz, S., Razzak, M. I., Akram, F., & Imran, M. (2020). A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning. *Circuits, Systems, and Signal Processing*, *39*(2), 757–775. https://doi.org/10.1007/s00034-019-01246-3

Abir, T. A., Siraji, J. A., & Ahmed, E. (2018). *Analysis of a novel MRI Based Brain Tumour Classification Using Probabilistic Neural Network (PNN). May*, 69–75

Badža, M. M., & Barjaktarović, M. C. (2020). Classification of brain tumors from mri images using a convolutional neural network. *Applied Sciences (Switzerland)*, *10*(6). https://doi.org/10.3390/app10061999

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*(1995), 60–88. https://doi.org/10.1016/j.media.2017.07.005

Chahal, P. K., Pandey, S., & Goel, S. (2020). A survey on brain tumor detection techniques for MR images. *Multimedia Tools and Applications*, *79*(29–30), 21771–21814. https://doi.org/10.1007/s11042-020-08898-3

Tandel, G. S., Biswas, M., Kakde, O. G., Tiwari, A., Suri, H. S., Turk, M., Laird, J. R., Asare, C. K., Ankrah, A. A., Khanna, N. N., Madhusudhan, B. K., Saba, L., & Suri, J. S. (2019). A review on a deep learning perspective in brain cancer classification. *Cancers*, *11*(1). https://doi.org/10.3390/cancers11010111

Noreen, N., Palaniappan, S., Qayyum, A., Ahmad, I., Imran, M., & Shoaib, M. (2020). A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor. *IEEE Access*, *8*, 55135–55144. https://doi.org/10.1109/ACCESS.2020.2978629

Deepak, S., & Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine*, *111*(March), 103345. https://doi.org/10.1016/j.compbiomed.2019.103345

Deepak, S., & Ameer, P. M. (2020). Automated Categorization of Brain Tumor from MRI

Using CNN features and SVM. *Journal of Ambient Intelligence and Humanized Computing*, *0123456789*. https://doi.org/10.1007/s12652-020-02568-w

Talo, M., Baloglu, U. B., Yıldırım, Ö., & Rajendra Acharya, U. (2019). Application of deep transfer learning for automated brain abnormality classification using MR images. *Cognitive Systems Research*, *54*, 176–188. https://doi.org/10.1016/j.cogsys.2018.12.007

Kaur, T., & Gandhi, T. K. (2020). Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications*, *31*(3). https://doi.org/10.1007/s00138-020-01069-2

Dunsheng Liu, Yuanning Liu, and L. D. (2019). *Improved Res-Net for brain tumor classification. pdf*. International Conference on Neural Information Processing.

Yiming Cheng, Guihe Qin, Rui Zhao, and Y. L. (2019). *Multi-input Capsule Network For Brain Tumor Classification. pdf*. DOI: 10.1007/978-3-030-36708-4_43.

Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., Yun, Z., Wang, Z., & Feng, Q. (2015). Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS ONE*, *10*(10), 1–13. https://doi.org/10.1371/journal.pone.0140381

Ismael, M. R., & Abdel-Qader, I. (2018). Brain Tumor Classification via Statistical Features and Back-Propagation Neural Network. *IEEE International Conference on Electro Information Technology*, *2018-May*, 252–257. https://doi.org/10.1109/EIT.2018.8500308

Afshar, P., Mohammadi, A., & Plataniotis, K. N. (2018). Brain Tumor Type Classification via Capsule Networks. *Proceedings - International Conference on Image Processing, ICIP*, 3129–3133. https://doi.org/10.1109/ICIP.2018.8451379

Pashaei, A., Sajedi, H., & Jazayeri, N. (2018). Brain tumor classification via convolutional neural network and extreme learning machines. *2018 8th International Conference on Computer and Knowledge Engineering, ICCKE 2018*, *Iccke*, 314–319. https://doi.org/10.1109/ICCKE.2018.8566571

Nyoman Abiwinanda, Muhammad Hanif, S. T. H., & Astri Handayani, and T. R. M. (2019). Brain Tumor Classification Using Convolutional Neural Network. *Springer*.

Kabir Anaraki, A., Ayati, M., & Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybernetics and Biomedical Engineering*, *39*(1), 63–74. https://doi.org/10.1016/j.bbe.2018.10.004

Swati, Z. N. K., Zhao, Q., Kabir, M., Ali, F., Ali, Z., Ahmed, S., & Lu, J. (2019). Brain tumor classification for MR images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*, *75*, 34–46. https://doi.org/10.1016/j.compmedimag.2019.05.001

Toğaçar, M., Ergen, B., & Cömert, Z. (2020). BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model. *Medical Hypotheses*, *134*(November 2019). https://doi.org/10.1016/j.mehy.2019.109531

Ghosal, P., Nandanwar, L., Kanchan, S., Bhadra, A., Chakraborty, J., & Nandi, D. (2019). Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. *2019 2nd International Conference on Advanced Computational and Communication Paradigms, ICACCP 2019*, 1–6. https://doi.org/10.1109/ICACCP.2019.8882973

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). Machine Learning Algorithms and Applications. In *CRC Press.*

Naqa, I. El, & Murphy, M. J. (2015). *Machine Learning in Radiation Oncology*. 3–11. https://doi.org/10.1007/978-3-319-18305-3

Mahalakshmi, D. M., & Sumathi, S. (2020). *Performance Analysis of Svm and Deep Learning With Cnn for Brain Tumor Detection and Classification*. *9102*(February), 2145–2152. https://doi.org/10.21917/ijivp.2020.0307

Gokulalakshmi, A., Karthik, S., Karthikeyan, N., & Kavitha, M. S. (2020). ICM-BTD: improved classification model for brain tumor diagnosis using discrete wavelet transform-based feature extraction and SVM classifier. *Soft Computing*, *24*(24), 18599–18609. https://doi.org/10.1007/s00500-020-05096-z

Tandel, G. S., Balestrieri, A., Jujaray, T., Khanna, N. N., Saba, L., & Suri, J. S. (2020). Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm. *Computers in Biology and Medicine*, *122*(May), 103804. https://doi.org/10.1016/j.compbiomed.2020.103804

Rikiya Yamashita, Mizuho Nishio, R. K. G. D. & K. T. (2018). REVIEW Convolutional neural networks: an overview and application in radiology. *Springer*.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Szegedy, S. I. and C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:448-456, 2015.*

Sajid, S., Hussain, S., & Sarwar, A. (2019). Brain Tumor Detection and Segmentation in MR Images Using Deep Learning. *Arabian Journal for Science and Engineering*, *44*(11), 9249–9261. https://doi.org/10.1007/s13369-019-03967-8

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, A. R. S. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15 (2014) 1929-195*.

Kurbiel, T., & Khaleghian, S. (2017). Training of Deep Neural Networks based on Distance Measures using RMSProp. *ArXiv*, 1–6.

Mzoughi, H., Njeh, I., Wali, A., Slima, M. Ben, BenHamida, A., Mhiri, C., & Mahfoudhe, K. Ben. (2020). Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification. *Journal of Digital Imaging*, *33*(4), 903–915. https://doi.org/10.1007/s10278-020-00347-9

Sun, K. H. X. Z. S. R. J. (2015). Deep Residual Learning for Image Recognition and 1-acetyloxycarbazole-2- carbaldehydes. *IEEE Explore*.

Khan, H. A., Jue, W., Mushtaq, M., & Mushtaq, M. U. (2020). Brain tumor classification in MRI image using convolutional neural network. *Mathematical Biosciences and Engineering*, *17*(5), 6203–6216. https://doi.org/10.3934/MBE.2020328

Talo, M. (2019). Convolutional neural networks for multi-class histopathology image classification. *ArXiv*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Hassan, M., Ali, S., Alquhayz, H., & Safdar, K. (2020). Developing intelligent medical image modality classification system using deep transfer learning and LDA. *Scientific Reports*, *10*(1), 1–14. https://doi.org/10.1038/s41598-020-69813-2

Wacker, J., Ladeira, M., & Nascimento, J. E. V. (2019). Transfer learning for brain tumor segmentation. *ArXiv*.

Mehrotra, R., Ansari, M. A., Agrawal, R., & Anand, R. S. (2020). A Transfer Learning approach for AI-based classification of brain tumors. *Machine Learning with Applications*, *2*(July), 100003. https://doi.org/10.1016/j.mlwa.2020.100003

Figshare brain tumor dataset. (2018). https://doi.org/. https://do.org/10.6084/m9.fgshare.1512427.

Işin, A., Direkoğlu, C., & Şah, M. (2016). Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods. *Procedia Computer Science*, *102*(August), 317–324. https://doi.org/10.1016/j.procs.2016.09.407

Mohsen, H., El-Dahshan, E. A., El-Horbaty, E. M., & Salem, A. M. (2017). Brain tumor type classification based on support vector machine in magnetic resonance images. *Annals Of "Dunarea De Jos" University Of Galati, Mathematics, Physics, Theoretical Mechanics, Fascicle II, Year IX (XL)*, *1*, 75–88.

https://figshare.com/articles/dataset/brain

APPENDIX I

```
import os
os. environ ["KERAS_BACKEND"] = "tensorflow"
kerasBKED = os. environ ["KERAS_BACKEND"]
print(kerasBKED)
import argparse
import sys
import numpy as np
#import hdf5storage
import cv2
import tensorflow as tf
import matplotlib. pyplot as plt
import pickle
import time
import datetime


import keras
from keras. models import load_model
from keras. datasets import cifar10
from keras. preprocessing. image import ImageDataGenerator
from keras. models import Sequential
from keras. layers import Dense, Dropout, Activation, Flatten
from keras. layers import Conv2D, MaxPooling2D
from keras. callbacks import EarlyStopping, ModelCheckpoint

from google. colab import drive
drive. mount('/content/drive')


data_dir = os. path. Join ('/Users/Pshtiwan Jabar/Desktop/Data’) #. Replace ('\\', '/')
files = os. listdir (data_dir)
labels = []
images = []
masks= []
dim=256
for i, file in enumerate (files, start=1):
    mat_file = hdf5storage.loadmat(os. path. join (data_dir, file)) ['cjdata’] [0]
    image = cv2.resize(mat_file [2], dsize= (dim, dim), interpolation=cv2.INTER_CUBIC)
    mask = cv2.resize(mat_file [4]. astype('uint8'), dsize= (dim, dim),
interpolation=cv2.INTER_CUBIC)
    labels. Append (int (mat_file [0]))
    images. Append (image)
    masks. append (mask. astype(bool))
labels = np. array (labels)
images = np. array(images)
masks = np. array(masks)
np. save (os. path. join (data_dir, 'Label.npy'), labels)
np. save (os. path. join (data_dir, 'Data.npy'), images)
np. save (os. path. join (data_dir, 'masks.npy'), masks)
print ('Label.npy, Data.npy, masks.npy saved in’, data_dir)
```

```
integer_to_class = {'1': 'meningioma (1)', '2': 'glioma (2)', '3': 'pituitary tumor (3)'}
plt. figure (figsize= (16, 8))
for i, idx in enumerate (np. random. randint (images. shape [0], size=12), start=1):
    plt. subplot (3, 6, i)
    plt. imshow(images[idx], cmap='Blues')

    # set black pixel as transparent for the mask
    mask = np.ma. masked_where(masks[idx] == False, masks[idx])
    plt. imshow (mask, alpha=0.4, cmap='spring')

    plt. title(integer_to_class[str(labels[idx])])
    plt. axis('off')
```

```
dim=256
from sklearn. model_selection import train_test_split
X = np. load ("/content/drive/My Drive/data3/Data.npy")
X=np.array(X)
Y = np. load ("/content/drive/My Drive/data3/Label.npy")
print (X. shape)
print (X. shape)
```

```
Y2 = ((Y. reshape (-1,1) & (2**np. arrange (3)))! = 0). astype(int)
X=Y.reshape(X. shape [0], dim, dim, 1)
X_train, X_test, Y_train, Y_test=train_test_split (X, Y2, random_state = 2)
def re_sample (X, Y):
    X_new=np. concatenate ((X, X, X))
    Y_new=np. concatenate ((Y, Y, Y))
    return X_new, Y_new
X_new, Y_new=re_sample (X, Y2)
print (X_new. shape)
print (Y_new. shape)


from keras. layers import Convolution2D, Dropout, Dense, MaxPooling2D
from keras. layers import BatchNormalization
from keras. layers import MaxPooling2D
from keras. layers import Flatten
from keras. models import Sequential
model=Sequential ()
model.add (BatchNormalization (input_shape = X_new. shape [1:]))
model.add (Convolution2D (32, (3,3), activation ='relu', input_shape = X_new. shape [1:]))
model.add(MaxPooling2D(pool_size=2))
model.add (Convolution2D (filters=64, kernel_size=4, padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add (Convolution2D (filters=128, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add (Convolution2D (filters=128, kernel_size=2, padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add (Dropout (0.25))
model.add (Flatten ())
model.add (Dense (units=128, activation = 'relu'))
model.add (Dense (units = 64, activation = 'relu'))
model.add (Dense (units = 32, activation = 'relu'))
model.add (Dense (units = 3, activation = 'softmax'))
print (model. Summary ())


model. compile (optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

score= [0,0]
T_accuracy= []
T_accuracy= []
T_loss= []
V_loss= []
history=model.fit (X_new, Y_new, epochs=30, validation_data= (X_test, Y_test), verbose = 1,
initial_epoch=0)
score = model. evaluate (X_test, Y_test, verbose=0)
v=float (history. history['accuracy’] [0])
t=float (history. history['val_accuracy’] [0])
T_accuracy. Append(v)
V_accuracy. append(t)
l=history. history['loss’] [0]
v_l=history. history['val_loss’] [0]
```

```
T_loss. append(l)
V_loss. append(v)


model. evaluate (X_test, Y_test)
plt. plot (history. history['accuracy'])
plt. plot (history. history['val_accuracy'])
plt. title ('model accuracy')
plt. ylabel('accuracy')
plt. xlabel('epoch')
plt. legend (['train', 'validation'], loc='upper left')
plt. show ()


plt. plot (history. history['loss'])
plt. plot (history. history['val_loss'])
plt. title ('model loss')
plt. ylabel('loss')
plt. xlabel('epoch')
plt. legend (['train', 'validation'], loc='upper left')
plt. show ()


Y_pred = model. predict(X_test)

Y_pred2= (Y_pred > 0.1)

Y_test3=np.zeros(len(Y_test))
for i in range(len(Y_test)):
    if Y_test[i][0] ==1:
       if Y_test[i][1] ==0:
          if Y_test[i][2] == 0:
               Y_test3[i]=1
    if Y_test[i][0] ==0:
        if Y_test[i][1] ==1:
            if Y_test[i][2] == 0:
               Y_test3[i]=2
    if Y_test[i][0] ==1:
        if Y_test[i][1] ==1:
            if Y_test[i][2] == 0:
               Y_test3[i]=3
```

```
Y_train3=np.zeros(len(Y_train))
for i in range(len(Y_test)):
    if Y_train[i][0] ==1:
       if Y_train[i][1] ==0:
          if Y_train[i][2] == 0:
               Y_train3[i]=1
    if Y_train[i][0] ==0:
        if Y_train[i][1] ==1:
            if Y_train[i][2] == 0:
               Y_train3[i]=2
    if Y_train[i][0] ==1:
        if Y_train[i][1] ==1:
            if Y_train[i][2] == 0:
               Y_train3[i]=3


Y_pred3=np.zeros(len(Y_pred2))
for i in range(len(Y_pred2)):
    if Y_pred2[i][0] ==True:
       if Y_pred2[i][1] ==False:
          if Y_pred2[i][2] == False:
              Y_pred3[i]=1
    if Y_pred2[i][0] ==False:
       if Y_pred2[i][1] ==True:
        if Y_pred2[i][2] == False:
             Y_pred3[i]=2
    if Y_pred2[i][0] ==True:
        if Y_pred2[i][1] ==True:
           if Y_pred2[i][2] == False:
            Y_pred3[i]=3


target_names = ['Meningioma', 'Glioma', 'Pituitary tumor']
from sklearn. metrics import classification_report
print (classification_report (Y_test3, Y_pred3, target_names=target_names, digits=4))


# Layers definitions
from keras import backend as K
for l in range (len (model. layers)):
    print (l, model. layers[l])
```

```
from keras. models import Model
model_feat = Model (inputs=model. layers [0]. input, outputs=model. layers [14]. output)


X_train, X_test, Y_train, Y_test=train_test_split (X, Y2, random_state=4)
feat train = model_feat. predict(X_train)
print (feat_train. shape)
feat_test = model_feat. predict(X_test)
print (feat_test. shape)


from sklearn.svm import SVC
from sklearn. model_selection import GridSearchCV
parameters = {'kernel’: ['rbf'],
             'C’: [1, 10, 100, 1000],
             'gamma’: [1e-3, 1e-5]}
clf = GridSearchCV (SVC (), parameters)
clf.fit (feat_train, np. argmax (Y_train, axis=1))
svmclf = clf. best_estimator_
svmclf.fit (feat_train, np. argmax (Y_train, axis=1))
y_testSVM = svmclf. predict(feat_test)
score1=svmclf.score(feat_train, np. argmax (Y_train, axis=1))
print ('Accuracy in Trian SVM-CNN-TL:', score1)
score2 = svmclf. score (feat_test, np. argmax (Y_test, axis=1))
print ('Accuracy in test SVM-CNN-TL:', score2)


from sklearn. multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC
mysvm=OneVsRestClassifier(LinearSVC(random_state=0)). fit (feat_train, Y_train)
Y_pred_mysvm = mysvm. predict(feat_test)


Y_pred_mysvm3=np.zeros(len(Y_pred_mysvm))
for i in range(len(Y_test)):
   if Y_pred_mysvm[i][0] ==1:
     if Y_pred_mysvm[i][1] ==0:
       if Y_pred_mysvm[i][2] == 0:
           Y_pred_mysvm3[i]=1
   if Y_pred_mysvm[i][0] ==0:
      if Y_pred_mysvm[i][1] ==1:
         if Y_pred_mysvm[i][2] == 0:
           Y_pred_mysvm3[i]=2
```

```
    if Y_pred_mysvm[i][0] ==1:
        if Y_pred_mysvm[i][1] ==1:
            if Y_pred_mysvm[i][2] == 0:
               Y_pred_mysvm3[i]=3
Y_pred2= (Y_pred > 0.01)
for i in range(len(Y_pred2)):
    if Y_pred2[i][0] ==True:
       if Y_pred2[i][1] ==False:
          if Y_pred2[i][2] == False:
              Y_pred_mysvm3[i]=1
    if Y_pred2[i][0] ==False:
       if Y_pred2[i][1] ==True:
        if Y_pred2[i][2] == False:
             Y_pred_mysvm3[i]=2
    if Y_pred2[i][0] ==True:
        if Y_pred2[i][1] ==True:
           if Y_pred2[i][2] == False:
            Y_pred_mysvm3[i]=3

target_names = ['Meningioma', 'Glioma', 'Pituitary tumor']
from sklearn. metrics import classification_report
print (classification_report (Y_test3, Y_pred_mysvm3, target_names=target_names, digits=4))

import itertools
import numpy as np
import matplotlib. pyplot as plt
from sklearn. metrics import confusion matrix
def plot_confusion_matrix (cm, title='Confusion matrix', cmap=plt.cm. Oranges):
    plt. imshow (cm, interpolation='nearest', cmap=cmap)
    plt. title(title)
    plt. colorbar ()
    tick_marks = np. arange (cm. shape [1])
    plt. xticks (tick_marks, rotation=45)
    ax = plt.gca ()
    ax.set_xticklabels ((ax.get_xticks () +1). astype(str))
    plt. yticks(tick_marks)

    thresh = cm.max () / 2.
    for i, j in itertools. product (range (cm. shape [0]), range (cm. shape [1])):
        plt. Text (j, i, format (cm [i, j], '.1f'),
                 horizontalalignment="center",
                 color="white" if cm [i, j] > thresh else "black")

    plt. tight layout ()
    plt. ylabel ('True label')
    plt. xlabel ('Predicted label')

cm = confusion matrix (Y_test3, Y_pred_mysvm3)
np.set_printoptions(precision=1)
print ('Confusion matrix, without normalization')
print(cm)
fig, ax = plt. subplots ()
plot_confusion_matrix(cm)
plt. show ()
```

```
def resnet50_model (classes=1000, *args, **kwargs):
    # Create an input layer
    input = keras. layers. Input (X_new. shape [1:])
    # Create output layers
    output = keras. layers. ZeroPadding2D (padding=3, name='padding_conv1’) (input)
    output = keras. layers. Conv2D (64, (7, 7), strides= (2, 2), use bias=False, name='conv1’)
(output)
    output = keras. layers. BatchNormalization (axis=3, epsilon=1e-5, name='bn_conv1’)
(output)
    output = keras. layers. Activation ('relu', name='conv1_relu’) (output)
    output = keras. layers. MaxPooling2D ((3, 3), strides= (2, 2), padding='same',
name='pool1’) (output)
    output = conv_block (output, 3, [64, 64, 256], stage=2, block='a', strides= (1, 1))
    output = identity_block (output, 3, [64, 64, 256], stage=2, block='b')
    output = identity_block (output, 3, [64, 64, 256], stage=2, block='c')
    output = conv_block (output, 3, [128, 128, 512], stage=3, block='a')
    output = identity_block (output, 3, [128, 128, 512], stage=3, block='b')
    output = identity_block (output, 3, [128, 128, 512], stage=3, block='c')
    output = identity_block (output, 3, [128, 128, 512], stage=3, block='d')
    output = conv_block (output, 3, [256, 256, 1024], stage=4, block='a')
    output = identity_block (output, 3, [256, 256, 1024], stage=4, block='b')
    output = identity_block (output, 3, [256, 256, 1024], stage=4, block='c')
    output = identity_block (output, 3, [256, 256, 1024], stage=4, block='d')
    output = identity_block (output, 3, [256, 256, 1024], stage=4, block='e')
    output = identity_block (output, 3, [256, 256, 1024], stage=4, block='f')
    output = conv_block (output, 3, [512, 512, 2048], stage=5, block='a')
    output = identity_block (output, 3, [512, 512, 2048], stage=5, block='b')
    output = identity_block (output, 3, [512, 512, 2048], stage=5, block='c')
    output = keras. layers. GlobalAveragePooling2D(name='pool5’) (output)
    output = keras. layers. Dense (classes, activation='softmax', name='fc1000’) (output)
    # Create a model from input layer and output layers
    model = keras. models. Model (inputs=input, outputs=output, *args, **kwargs)
    # Print model
    print ()
    print (model. summary (), '\n')
    # Return a model
    return model
# Create an identity block
def identity_block (input, kernel_size, filters, stage, block):

    # Variables
    filters1, filters2, filters3 = filters
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'
    # Create layers
    output = keras. layers. Conv2D (filters1, (1, 1), kernel initializer='he_normal',
name=conv_name_base + '2a’) (input)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2a’) (output)
    output = keras. layers. Activation('relu’) (output)
    output = keras. layers. Conv2D (filters2, kernel_size, padding='same',
kernel_initializer='he_normal', name=conv_name_base + '2b’) (output)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2b’) (output)
    output = keras. layers. Activation('relu’) (output)
    output = keras. layers. Conv2D (filters3, (1, 1), kernel_initializer='he_normal',
name=conv_name_base + '2c’) (output)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2c’) (output)
    output = keras. layers. Add ([output, input])
    output = keras. layers. Activation('relu’) (output)
    # Return a block
    return output
# Create a convolution block
def conv_block (input, kernel_size, filters, stage, block, strides= (2, 2)):
    # Variables
    filters1, filters2, filters3 = filters
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'
    # Create block layers
    output = keras. layers. Conv2D (filters1, (1, 1), strides=strides,
kernel_initializer='he_normal', name=conv_name_base + '2a’) (input)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2a’) (output)
    output = keras. layers. Activation('relu’) (output)
    output = keras. layers. Conv2D (filters2, kernel_size, padding='same',
kernel_initializer='he_normal', name=conv_name_base + '2b’) (output)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2b’) (output)
    output = keras. layers. Activation('relu’) (output)
    output = keras. layers. Conv2D (filters3, (1, 1), kernel_initializer='he_normal',
name=conv_name_base + '2c’) (output)
    output = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '2c’) (output)
    shortcut = keras. layers. Conv2D (filters3, (1, 1), strides=strides,
kernel_initializer='he_normal', name=conv_name_base + '1’) (input)
    shortcut = keras. layers. BatchNormalization (axis=3, name=bn_name_base + '1’) (shortcut)
    output = keras. layers. Add ([output, shortcut])
    output = keras. layers. Activation('relu’) (output)
    # Return a block
    return output
model = resnet50_model (3)
```

ETHICAL APROVAL DOCUMENT

Date: _12__/_4__/_2021___

To the **Graduate School of Applied Sciences**

The research project titled “Deep transfer learning for brain tumor classification using pre-trained models, fine-tuning and support vector machine” has been evaluated. Since the researcher(s) will not collect primary data from humans, animals, plants or earth, this project does not need to go through the ethics committee.

Title: Assoc Prof Dr

Name Surname: Melike Sah Direkoglu

Signature:

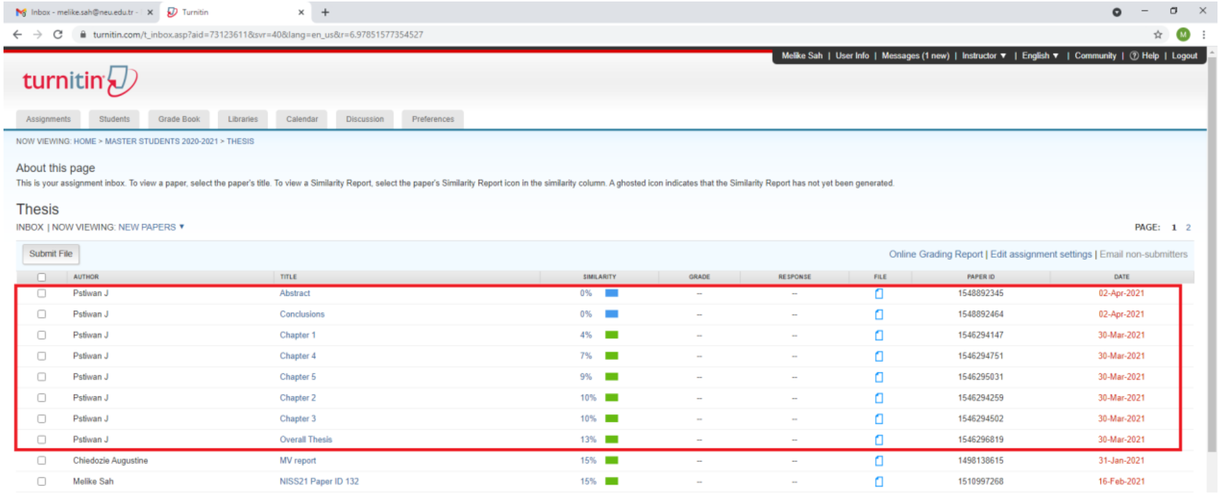
Role in the Research Project: Supervisor

APPENDIX II

Similarity Report

Chapters	Percentages
Abstract.doc/docx	0%
Chapter 1.doc/docx	4%
Chapter 2.doc/docx	10%
Chapter 3.doc/docx	10%
Chapter 4.doc/docx	7%
Chapter 5.doc/docx	9%
Conclusions.doc/docx	0%
*All.doc/docx	13%

*All.doc/docx document must include all your thesis chapters (except cover page, table of contents, acknowledge, declaration, references, appendix, list of figures, list of tables, and abbreviations list).


Inbox - melike.sah@neu.edu.tr -
Turnitin
turnitin.com/t_inbox.asp?aid=73123611&svr=40&lang=en_us&r=6.97851577354527
Melike Sah | User Info | Messages (1 new) | Instructor ▼ | English ▼ | Community | Help | Logout
turnitin
Assignments
Students
Grade Book
Libraries
Calendar
Discussion
Preferences

NOW VIEWING: HOME > MASTER STUDENTS 2020-2021 > THESIS

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

Thesis

INBOX | NOW VIEWING: NEW PAPERS ▼

PAGE: 1 2

Submit File

Online Grading Report | Edit assignment settings | Email non-submitters

□	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
□	Pstiwan J	Abstract	0%	--	--		1548892345	02-Apr-2021
□	Pstiwan J	Conclusions	0%	--	--		1548892464	02-Apr-2021
□	Pstiwan J	Chapter 1	4%	--	--		1546294147	30-Mar-2021
□	Pstiwan J	Chapter 4	7%	--	--		1546294751	30-Mar-2021
□	Pstiwan J	Chapter 5	9%	--	--		1546295031	30-Mar-2021
□	Pstiwan J	Chapter 2	10%	--	--		1546294259	30-Mar-2021
□	Pstiwan J	Chapter 3	10%	--	--		1546294502	30-Mar-2021
□	Pstiwan J	Overall Thesis	13%	--	--		1546296819	30-Mar-2021
□	Chiedozie Augustine	MV report	15%	--	--		1498138615	31-Jan-2021
□	Melike Sah	NISS21 Paper ID 132	15%	--	--		1510997268	16-Feb-2021

Regards,

Assoc. Prof Dr Melike Sah Direkoglu.